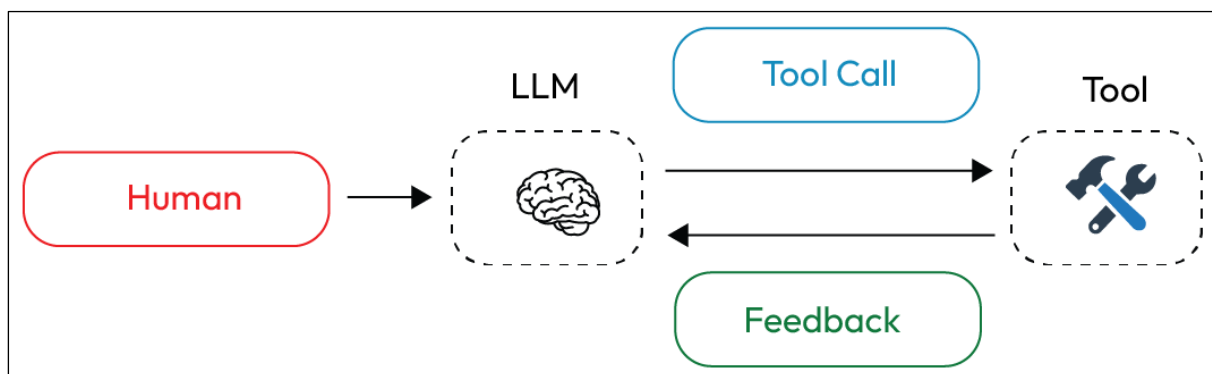
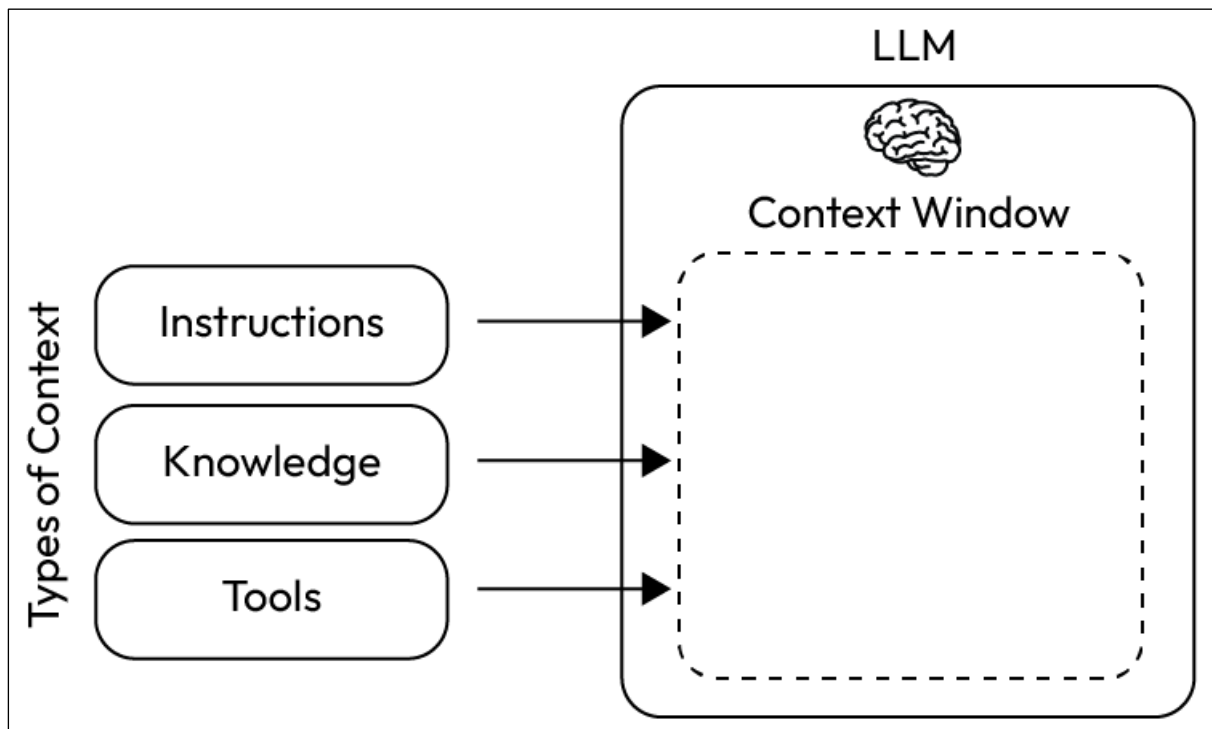
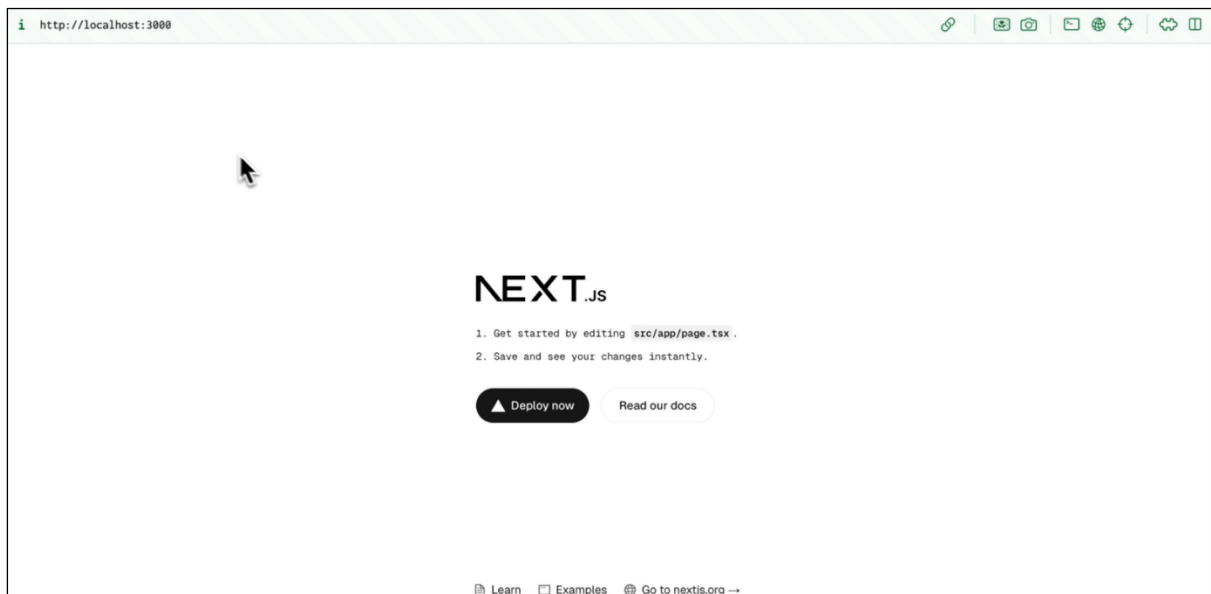


Chapter 1: Context Engineering



Chapter 2: The Gist of Claude Code



1

```
1+ # CLAUDE.md
2+
3+ This file provides guidance to Claude Code (claude.ai/code) when
4+ working with code in this repository.
5+
6+ ## Project Overview
7+ HookHub is a Next.js 15.4.6 application using the App Router
8+ architecture with TypeScript and Tailwind CSS.
9+
10+ ## Essential Commands
11+ ```bash
12+ # Development
13+ npm run dev          # Start development server on http://localhost:3000
14+
15+ # Production
16+ npm run build         # Create production build
17+ npm run start        # Start production server
18+
19+ # Code Quality
20+ npm run lint         # Run ESLint
21+ ```
22+
23+ ## Architecture
24+
25+ ...
```

```
Claude Code X
edenmarco@Edens-MacBook-Pro-2 claude-code-crash-course % claude

* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/edenmarco/GithubProjects/claude-code-crash-course

> /mcp
  (no content)

> /exit
  (no content)

>

? for shortcuts                                1 MCP server failed • /mcp for info
                                                ○ IDE disconnected
edenmarco@Edens-MacBook-Pro-2 claude-code-crash-course % claude mcp add playwright npx @p
laywright/mcp@latest -
```

⌘K to generate a command

Manage MCP servers

1. **context7** ✓ connected • Enter to view details
- > 2. **playwright** ✓ connected • Enter to view details

MCP Config locations (by scope):

- User config (available in all your projects):
 - /Users/edenmarco/.claude.json
- Project config (shared via .mcp.json):
 - /Users/edenmarco/GithubProjects/claude-code-crash-course/.mcp.json (file does not exist)
- Local config (private to you in this project):
 - /Users/edenmarco/.claude.json [project: /Users/edenmarco/GithubProjects/claude-code-crash-course]

For help configuring MCP servers, see:

<https://docs.anthropic.com/en/docs/claude-code/mcp>

Tool use

playwright - Navigate to a URL(url: "https://www.cnn.com") (MCP)
Navigate to a URL

Do you want to proceed?

> 1. Yes

2. Yes, and don't ask again for playwright - Navigate to a URL commands in /Users/edenmarco/GithubProjects/claude-code-crash-course
3. No, and tell Claude what to do differently (esc)

The screenshot shows the cursor.directory website. On the left is a sidebar with a list of technologies and their counts: TypeScript (23), Python (16), Next.js (12), React (12), PHP (8), JavaScript (6), TailwindCSS (5), Laravel (5), C# (4), Web Development (4), Game Development (4), Expo (4), React Native (4), Flutter (4), Tailwind (4), Testing (4), Vite (4), and Sucabase (4). The main content area is titled 'Next.js' and features three columns of AI prompts. The first column contains a prompt for a Senior Front-End Developer. The second column contains a prompt for an expert developer proficient in TypeScript, React, and Next.js, followed by a section titled 'Code Style and Structure' with bullet points. The third column features a Sentry advertisement with the text 'The mid monitoring tool that fixes the code you prompted' and 'Code breaks. Fix it faster with Sentry'. At the bottom right, there is a BrainGrid AI logo.

> what r u expert in?

- I'm an expert AI assistant specialized in software engineering and coding tasks. I can help with:

- **Full-stack development** - Frontend (React, Vue, Angular), Backend (Node.js, Python, Go, Rust, etc.)
- **Debugging and troubleshooting** - Finding and fixing bugs, performance issues, type errors
- **Code refactoring** - Improving code structure, readability, and maintainability
- **Building features** - Implementing new functionality from scratch or extending existing code
- **Testing** - Writing unit tests, integration tests, debugging test failures
- **DevOps** - CI/CD, Docker, deployment configurations, build processes
- **Database work** - SQL, NoSQL, migrations, query optimization
- **API development** - REST, GraphQL, WebSockets
- **Security analysis** - Defensive security, vulnerability detection, security best practices
- **Code reviews** - Analyzing code quality, suggesting improvements
- **Documentation** - When requested, creating technical docs and API documentation

I have access to your file system and can read, write, and execute code directly. I also have web browsing capabilities and can fetch up-to-date library documentation through Context7.

Ready to code?

Here is Claude's plan:

Plan for HookHub MVP Specification

I'll create a comprehensive specification document for HookHub that includes:

1. **Project Overview** - Clear definition of what HookHub is and its purpose
2. **MVP Scope** - Focus on displaying hooks only (no submission/editing features yet)
3. **Data Model** - Structure for hook information (name, category, description, repo link)
4. **UI/UX Requirements** - Grid layout for browsing hooks
5. **Technical Architecture** - Simple tech stack recommendations
6. **User Stories** - Basic user interactions for MVP
7. **Future Considerations** - Potential features for post-MVP

The spec will be written in Markdown format and saved as `hookhub/SPEC.md`

Would you like to proceed?

1. Yes, and auto-accept edits
2. Yes, and manually approve edits
3. No, keep planning



The screenshot shows a web browser window with the following content:

- Page Title:** hookhub
- Document Title:** # HookHub - Product Specification
- Section 1: Project Overview**

HookHub is a community-driven platform for discovering, browsing, and sharing open-source Claude Code hooks. It serves as a centralized repository where developers can find pre-built hooks to enhance their Claude Code workflows, improve productivity, and implement best practices.
- Section 2: Vision**

To become the go-to resource for Claude Code users seeking hook implementations, fostering a collaborative ecosystem where developers share their custom hooks and workflows.
- Section 3: Target Audience**
 - Claude Code users looking for ready-to-use hooks
 - Developers wanting to share their hook implementations
 - Teams seeking standardized Claude Code workflows
 - Users new to Claude Code hooks wanting to learn from examples
- Section 4: MVP Scope**
- Section 5: In Scope (MVP)**
 - **Display-only functionality** - Browse and view existing hooks
 - **Grid-based hook gallery** - Visual presentation of available hooks
 - **Basic categorization** - Organize hooks by type/purpose
 - **GitHub integration** - Link to source repositories

The browser window also shows a sidebar with the file structure of the project, including files like `settings.local.json`, `hookhub`, `public`, `src/app`, `favicon.ico`, `globals.css`, `layout.tsx`, `page.tsx`, `.gitignore`, `CLAUDE.md`, `eslint.config.mjs`, `next-env.d.ts`, `next.config.ts`, `package-lock.json`, `package.json`, `postcss.config...`, `README.md`, `tsconfig.json`, `memory/frontend`, `CLAUD.MD`, and `spec`.

The bottom of the browser window shows a terminal window with the following output:

```
Next.js 15.4.6
- Local:    http://localhost:3000
- Network:  http://192.168.68.106:3000

✓ Starting...
✓ Ready in 1367ms
○ Compiling / ...
✓ Compiled / in 2.4s (612 modules)
GET / 200 in 2625ms
```

Claude Code hookhub X

• Now let me run the linting command to ensure the code is clean:

• Bash(npm run lint)

> hookhub@0.1.0 lint

> next lint

~ +4 lines (ctrl+r to expand)

• Perfect! The main page grid has been implemented according to the specifications in the CLAUDE.md file. The implementation includes:

✓ Complete Hook type system with all enums and interfaces

✓ Sample hook data with 12 curated hooks including the featured ones from the spec

✓ Responsive grid layout (1 column mobile, 2 tablet, 3-4 desktop)

✓ HookCard component with all required elements:

- Hook name, category badges, description

- Author, GitHub stars, language indicators

- Hook type tags, "View on GitHub" button

- Color-coded categories and language indicators

✓ Header with branding and search bar

✓ Hero section with explanatory text

✓ Featured hooks section and regular hooks section

✓ Dark mode support throughout

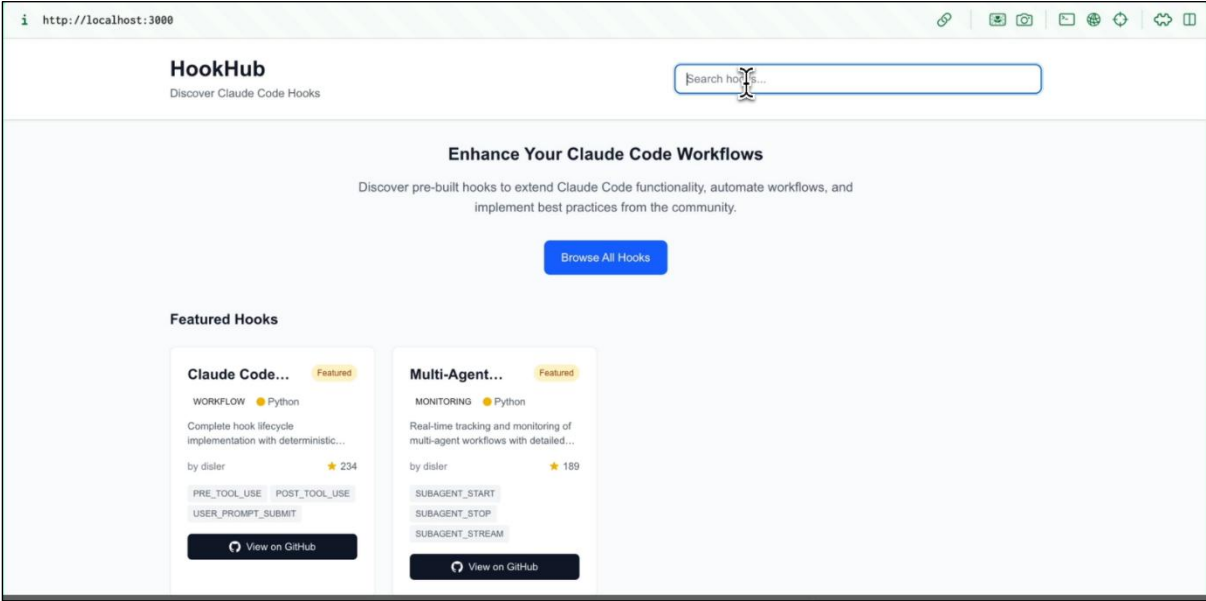
✓ Fully responsive design with mobile-first approach

The grid layout follows the spec exactly with proper responsive breakpoints and all the card components include the required information like category badges, GitHub stars, language indicators, and hook types.

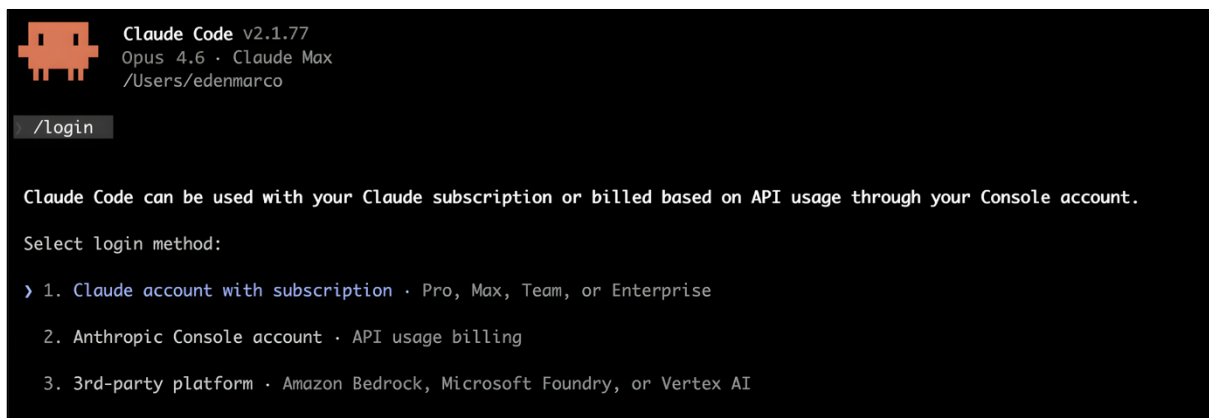
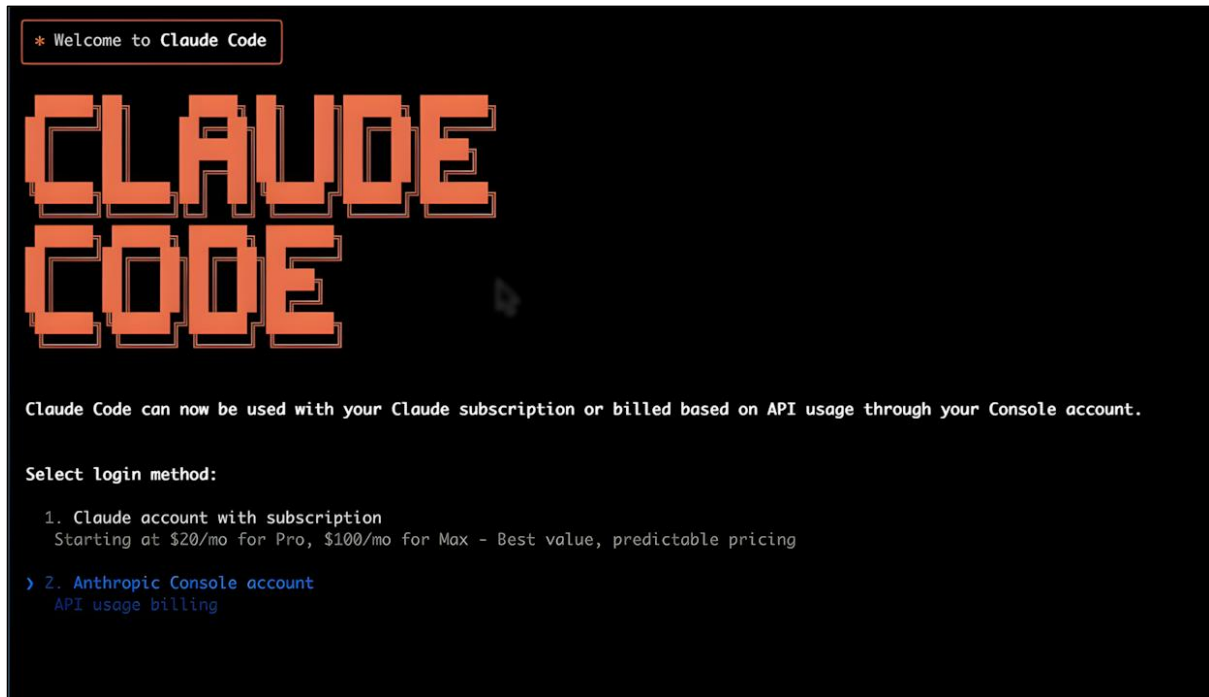
> █

•• auto-accept edits on (shift+tab to cycle)

In page.tsx



Chapter 3: Getting Started with Claude Code – A Tour of Essential Commands



* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/edenmarco/Desktop/planning

Tips for getting started:

1. Ask Claude to create a new app or clone a repository
2. Use Claude to help with file analysis, editing, bash commands and git
3. Be as specific as you would with another engineer for the best results
4. ✓ Run /terminal-setup to set up terminal integration

/cost

```
L Total cost:           $0.0000
  Total duration (API):  0s
  Total duration (wall): 31.3s
  Total code changes:    0 lines added, 0 lines removed
  Usage:                 0 input, 0 output, 0 cache read, 0 cache write
```

Tips for getting started:

1. Ask Claude to create a new app or clone a repository
2. Use Claude to help with file analysis, editing, bash commands and git
3. Be as specific as you would with another engineer for the best results
4. ✓ Run /terminal-setup to set up terminal integration

> /cost

```
L Total cost:           $0.0000
  Total duration (API):  0s
  Total duration (wall): 31.3s
  Total code changes:    0 lines added, 0 lines removed
  Usage:                 0 input, 0 output, 0 cache read, 0 cache write
```

> hello!

• Hello! How can I help you today?

> /cost

```
L Total cost:           $0.2873
  Total duration (API):  3.9s
  Total duration (wall): 40.3s
  Total code changes:    0 lines added, 0 lines removed
  Usage by model:
    claude-3-5-haiku:    87 input, 27 output, 0 cache read, 0 cache write
    claude-opus:         3 input, 13 output, 0 cache read, 15.3k cache write
```

> /usage

Settings: Status Config **Usage** (←/→ or tab to cycle)

Current session

 0% used
Resets 2pm (Asia/Jerusalem)

Current week (all models)

 0% used
Resets Jan 24, 11am (Asia/Jerusalem)

Current week (Sonnet only)

 0% used

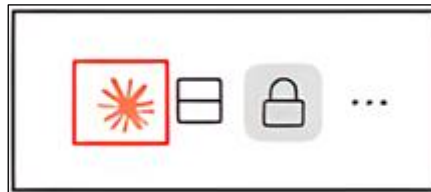
> /

/add-dir	Add a new working directory
/agents	Manage agent configurations
/bug	Submit feedback about Claude Code
/clear (reset)	Clear conversation history and free up context
/compact	Clear conversation history but keep a summary in context. Optional
/config (theme)	Open config panel
/cost	Show the total cost and duration of the current session
/doctor	Diagnose and verify your Claude Code installation and settings
/exit (quit)	Exit the REPL

Settings

Configure Claude Code preferences

> Auto-compact	true
Use todo list	true
Checkpointing	true
Verbose output	false
Auto-updates	true
Theme	Dark mode
Notifications	Auto
Editor mode	normal
Model	Default (recommended)
Auto-connect to IDE	false



Hook Configuration

Hooks are shell commands you can register to run during Claude Code processing. [Docs](#)

- Each hook event has its own input and output behavior
- Multiple hooks can be registered per event, executed in parallel
- Any changes to hooks outside of /hooks require a restart
- Timeout: 60 seconds

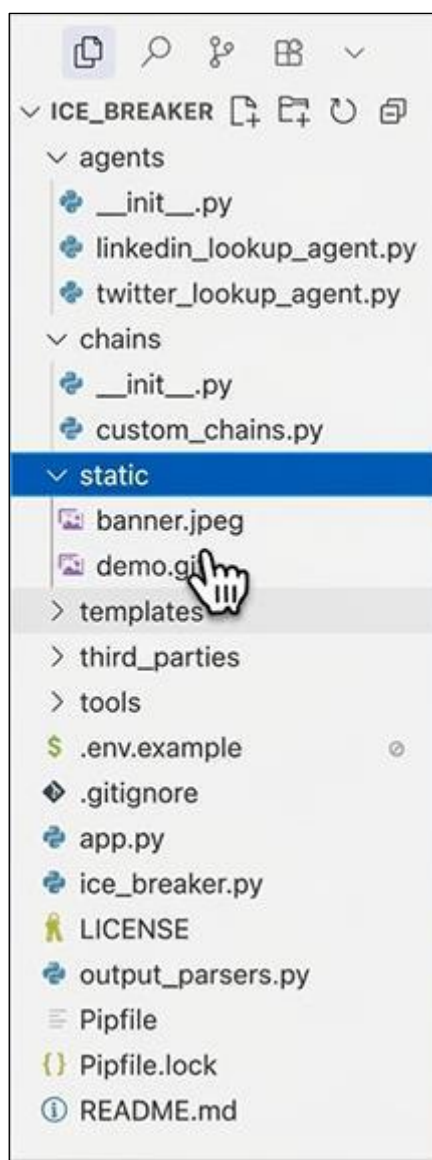
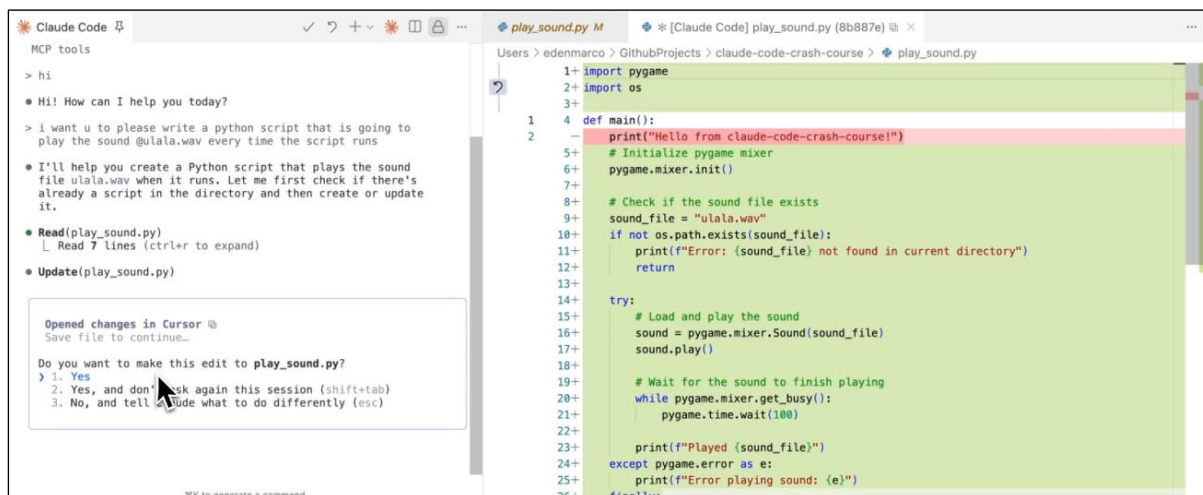
▲ CRITICAL SECURITY WARNING - USE AT YOUR OWN RISK

Hooks execute arbitrary shell commands with YOUR full user permissions without confirmation.

- You are SOLELY RESPONSIBLE for ensuring your hooks are safe and secure
- Hooks can modify, delete, or access ANY files your user account can access
- Malicious or poorly written hooks can cause irreversible data loss or system damage
- Anthropic provides NO WARRANTY and assumes NO LIABILITY for any damages resulting from hook usage
- Only use hooks from trusted sources to prevent data exfiltration
- Review [the hooks documentation](#) before proceeding

Select hook event:

- > 1. PreToolUse - Before tool execution
- 2. PostToolUse - After tool execution
- 3. Notification - When notifications are sent
- 4. UserPromptSubmit - When the user submits a prompt
- ↓ 5. Stop - Right before Claude concludes its response

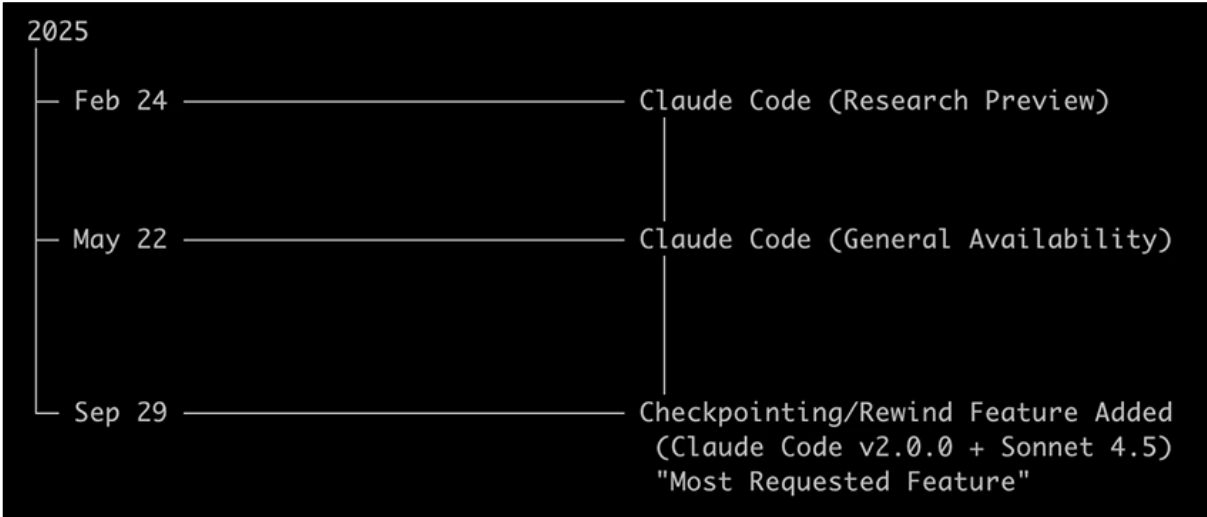


☀ Claude Code 📌

- ☐ Understand code architecture and main components
- ☐ Create comprehensive CLAUDE.md file
- ☐ Check for existing CLAUDE.md, README.md, and cursor/copilot rules
- **Read(Pipfile)**
 - └ Read 24 lines (ctrl+r to expand)
- **Search(pattern: "CLAUDE.md")**
 - └ Found 0 files
- **Search(pattern: "*.cursorrules")**
 - └ Found 0 files
- **Search(pattern: ".cursor/**/*")**
 - └ Found 0 files
- **Search(pattern: ".github/copilot-instructions.md")**
 - └ Found 0 files
- **Update Todos**
 - └ ☒ ~~Explore repository structure and key files~~
 - └ ☒ ~~Analyze build/test/lint commands from package.json or other config files~~
 - └ ☒ ~~Check for existing CLAUDE.md, README.md, and cursor/copilot rules~~
 - └ ☐ Understand code architecture and main components
 - └ ☐ Create comprehensive CLAUDE.md file
- **Read(app.py)**
 - └ Read 36 lines (ctrl+r to expand)
- **Read(ice_breaker.py)**
 - └ Read 52 lines (ctrl+r to expand)
- **Read(agents/linkedin_lookup_agent.py)**
 - └ Read 47 lines (ctrl+r to expand)

```
CLAUDE.md X
C: > Users > > Desktop > IceBreaker-main > IceBreaker-main > CLAUDE.md > abc # CLAUDE.md

1  # CLAUDE.md
9  ## Development Commands
23
24  ### Code Quality Tools
25  ```bash
26  # Format code with Black
27  pipenv run black .
28
29  # Sort imports with isort
30  pipenv run isort .
31
32  # Lint code with pylint
33  pipenv run pylint [module_name]
34  ```
35
36  ### Testing
37  ```bash
38  # Note: No test files currently exist, but the README mentions pytest
39  pipenv run pytest .
40  ```
41
42  ## Architecture & Key Components
43
44  ### Core Pipeline Flow
45  1. **Web Interface** (`app.py`): Flask server handling UI and API endpoints
46  2. **Main Logic** (`ice_breaker.py`): Orchestrates the entire ice breaker generation pipeline
47  3. **Agent System**: Profile lookup agents for LinkedIn and Twitter discovery
48  4. **Data Extraction**: Third-party integrations for scraping social media data
49  5. **AI Processing**: LangChain chains for generating summaries, interests, and ice breakers
50  6. **Output Parsing**: Structured response formatting using Pydantic models
51
52  ### Key Modules
```



HooksHub

Publish & Share Claude Code Hooks

Discover, share, and publish powerful hooks for Claude Code. Extend your workflow with community-driven automation and custom integrations.

Browse Hooks

Publish Hook



Discover

Find hooks for any workflow



Share

Publish your custom hooks



Integrate

Install with one command

Claude Code

Rewind

Restore the code and/or conversation to the point before...

how do i bootstrap an nextjs project?
No code changes

add a landing page for hookshub to publish claude code hooks. make it cool and rainy
page.tsx +117 -95

now add an FAQ section
page.tsx +62 -0

add a footer
page.tsx +66 -0

> /rewind
No code changes

⌘K to generate a command

Problems

Output

Debug Console

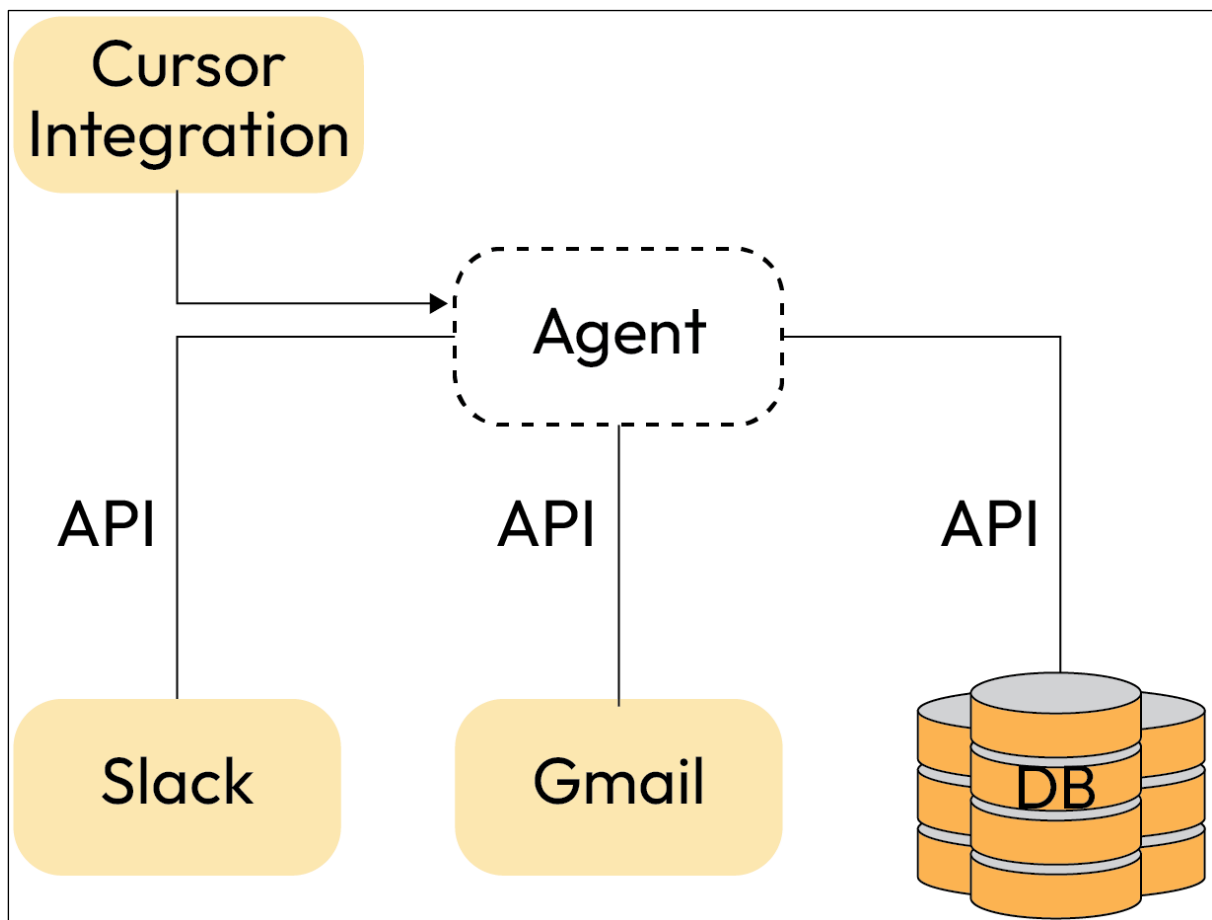
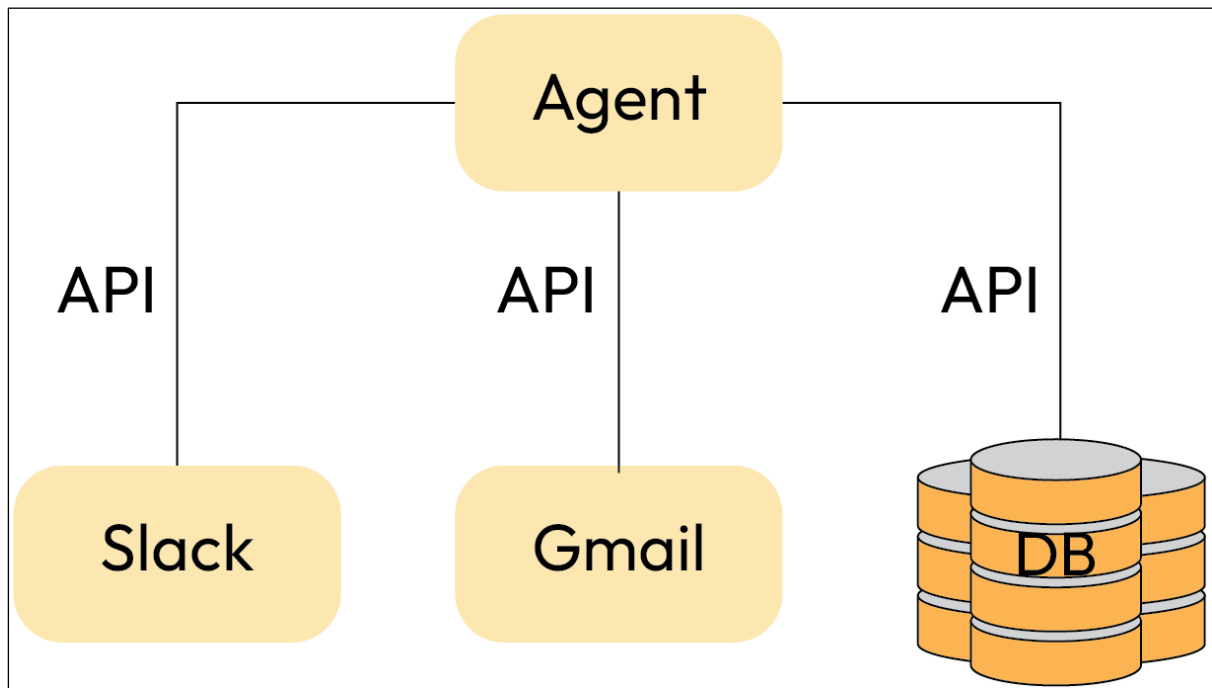
Terminal

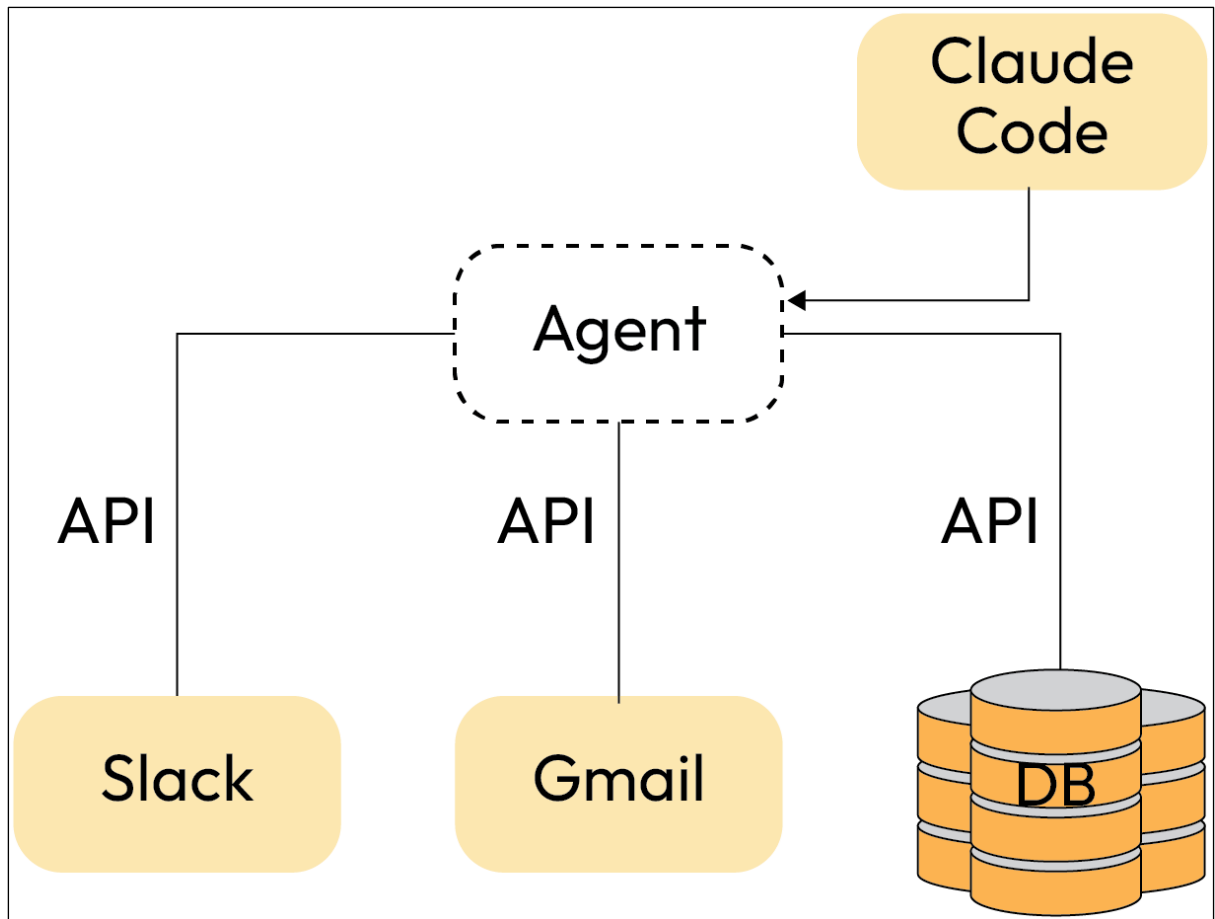
Ports

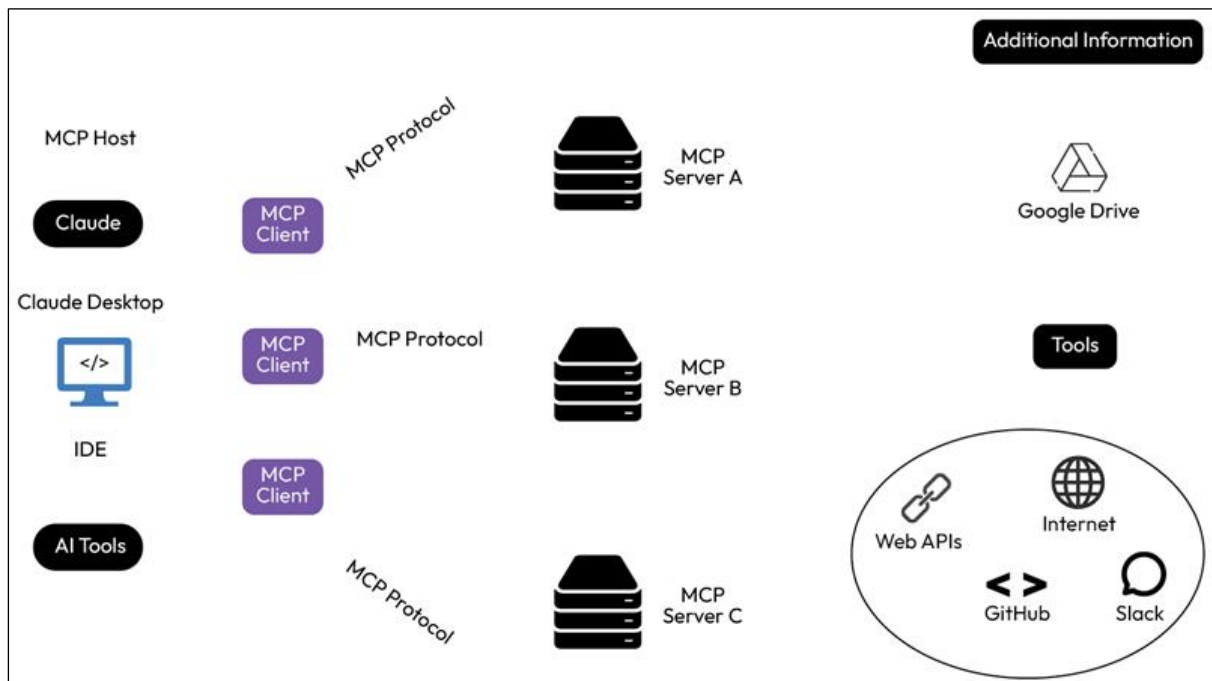
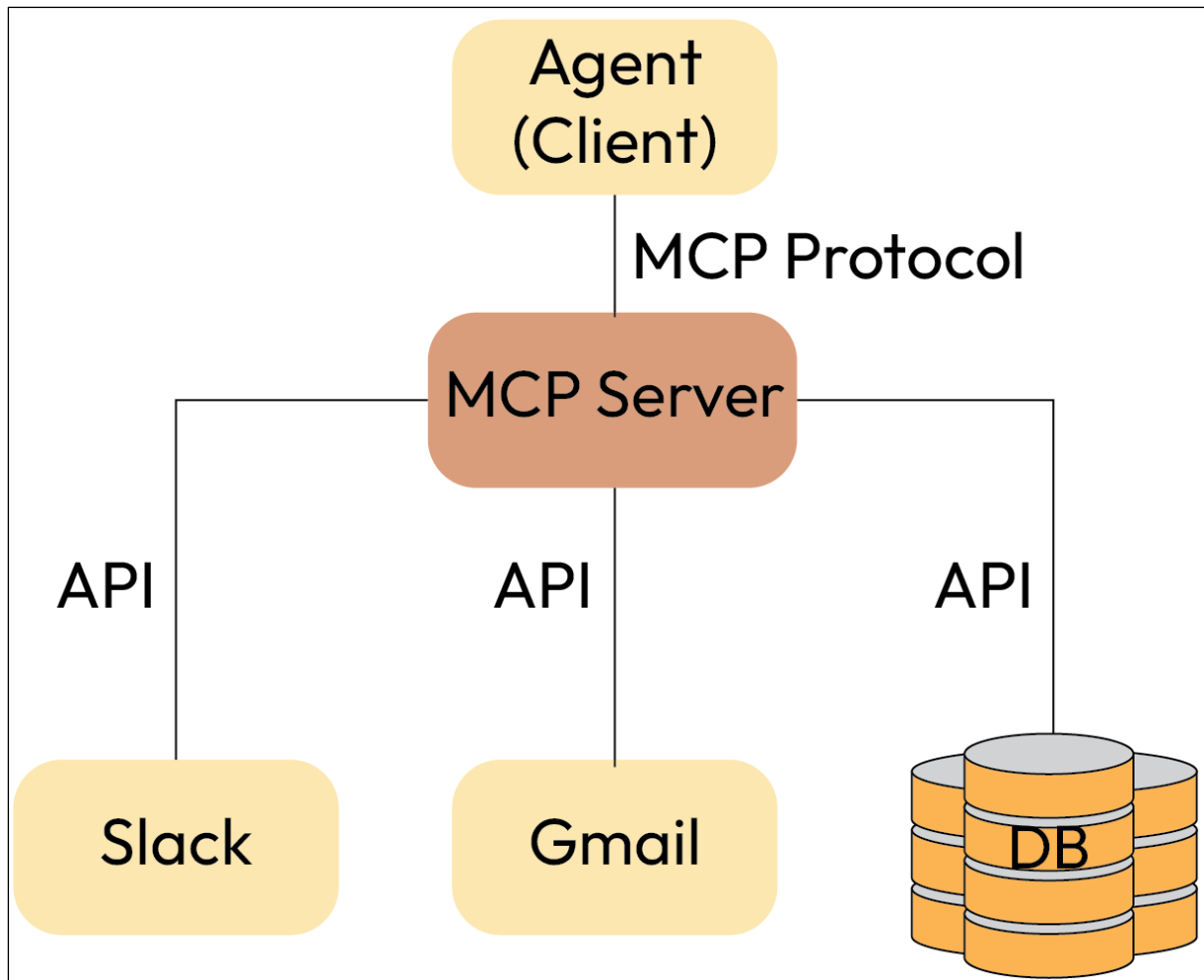
node - my-app

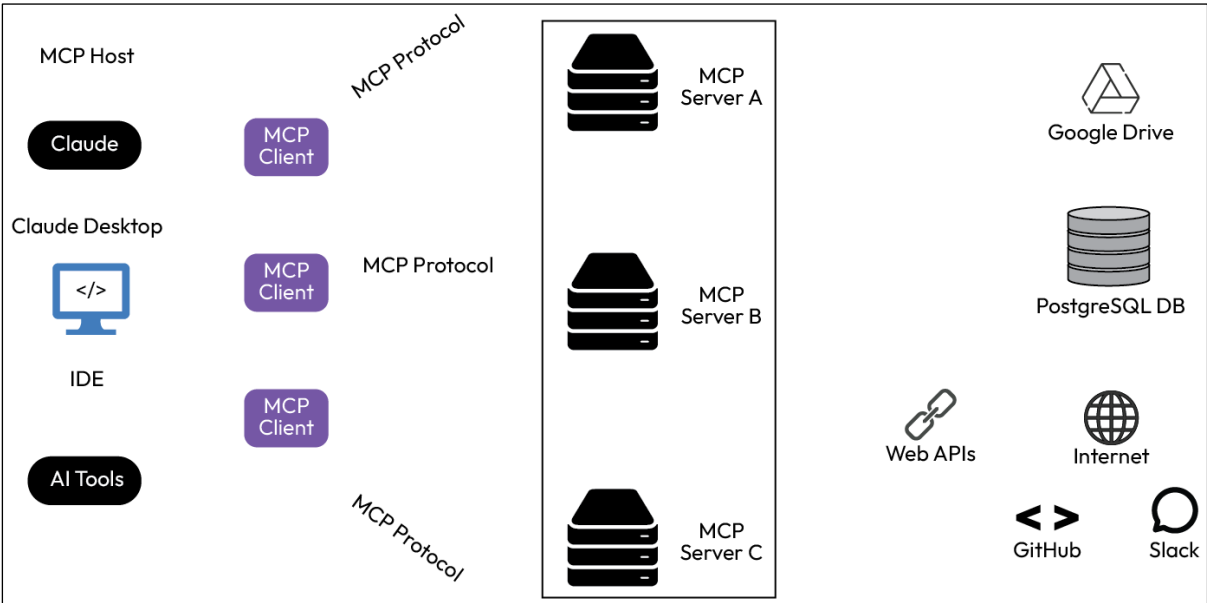
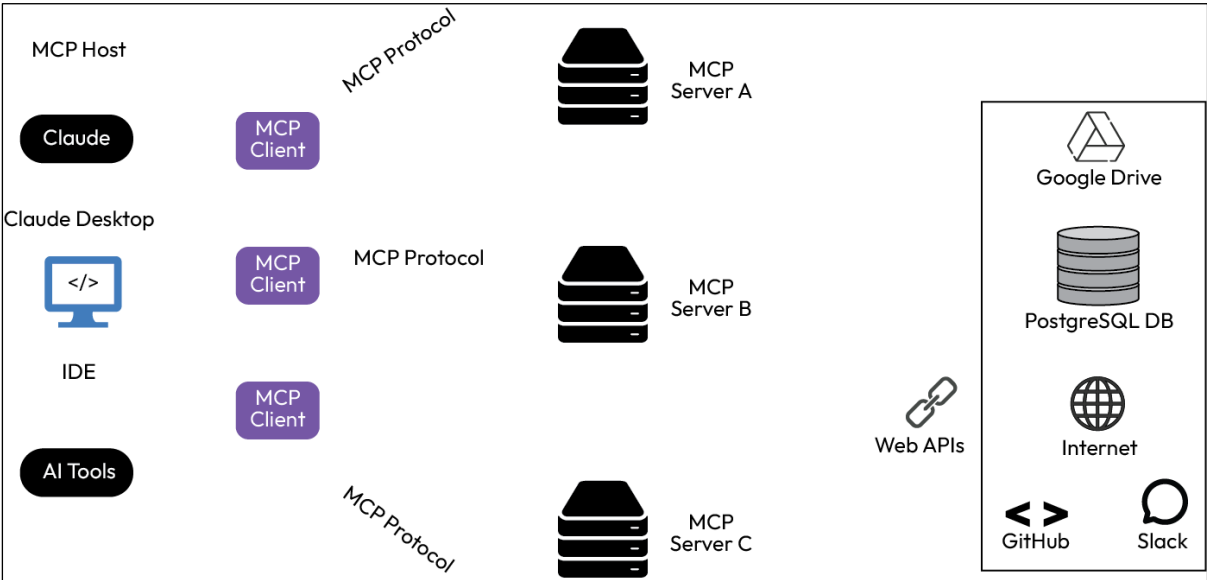
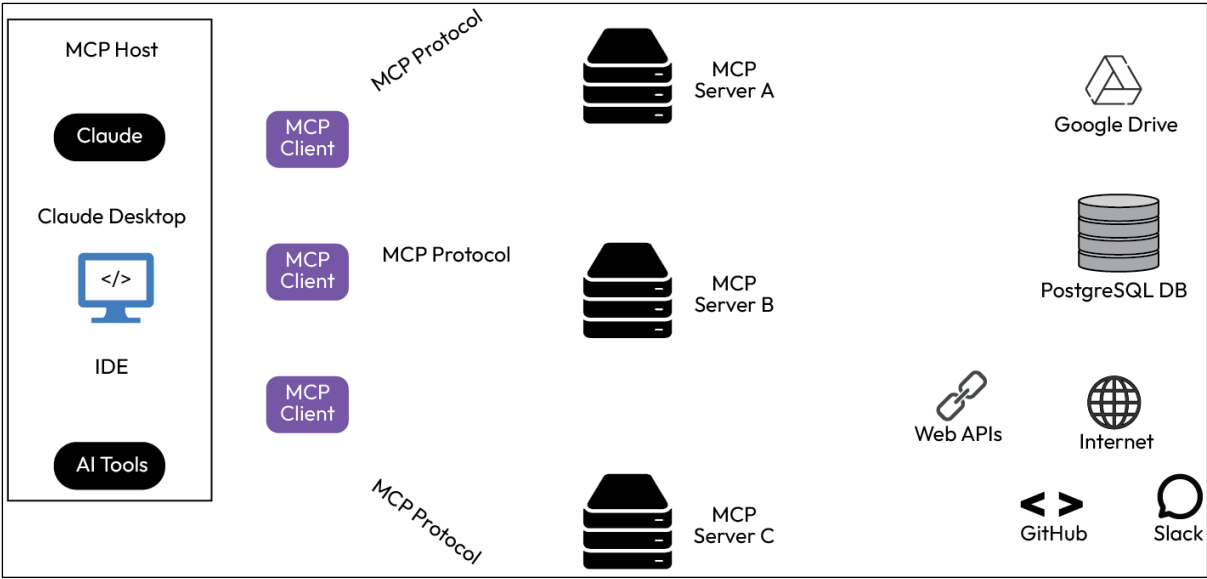
```
GET / 200 in 2501ms
✓ Compiled in 274ms (577 modules)
GET / 200 in 99ms
✓ Compiled in 106ms (577 modules)
GET / 200 in 56ms
✓ Compiled in 130ms (577 modules)
GET / 200 in 44ms
✓ Compiled in 139ms (577 modules)
GET / 200 in 53ms
```

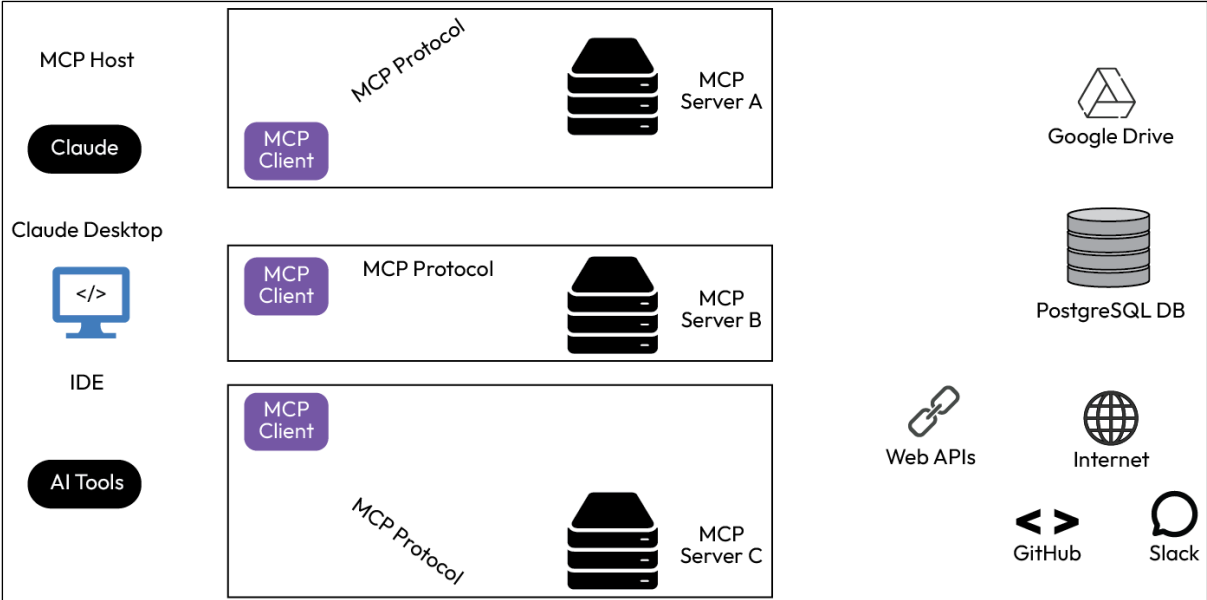
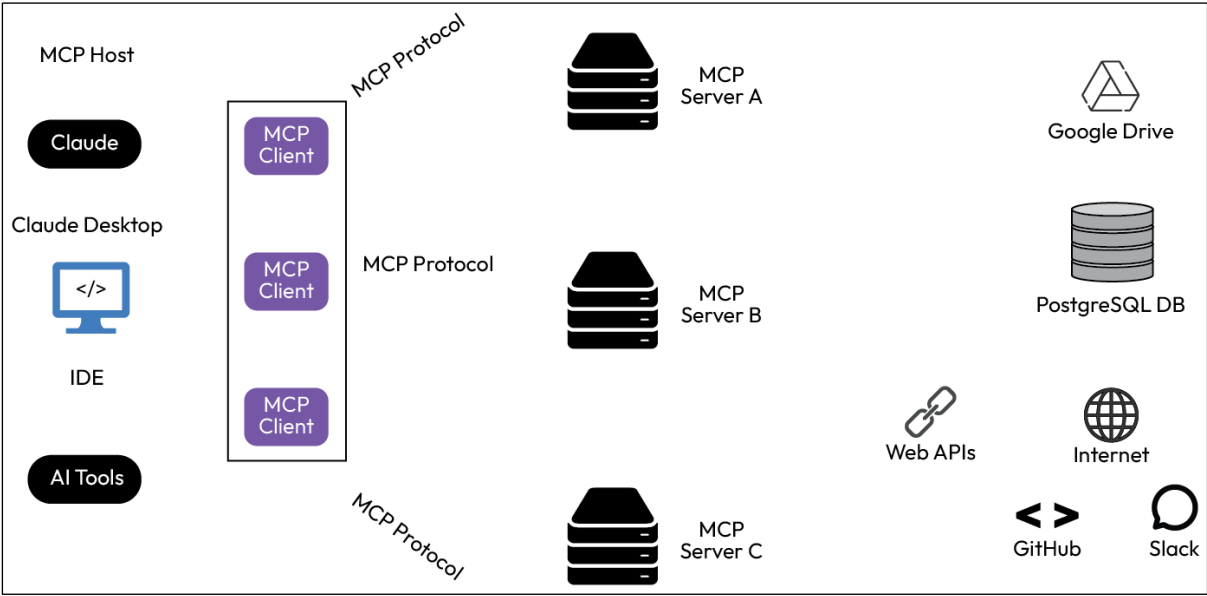
Chapter 4: Extending Claude Code with MCP Servers and Plugins

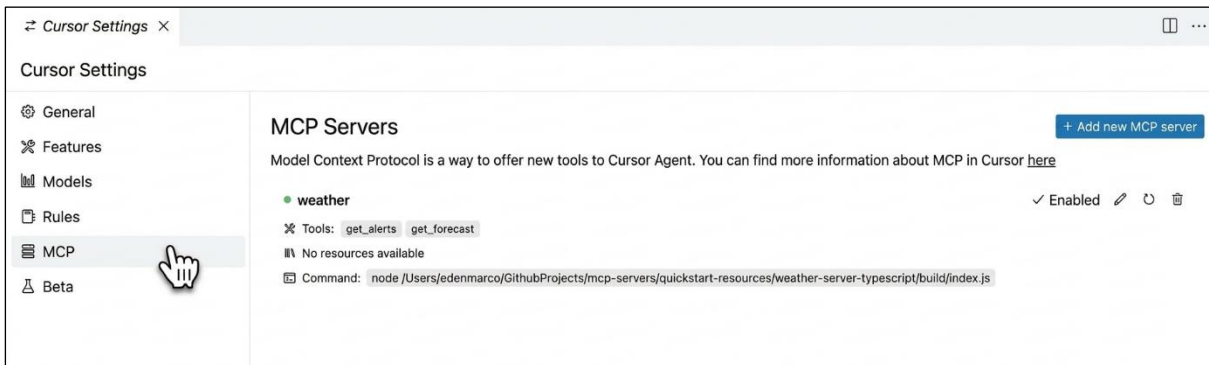
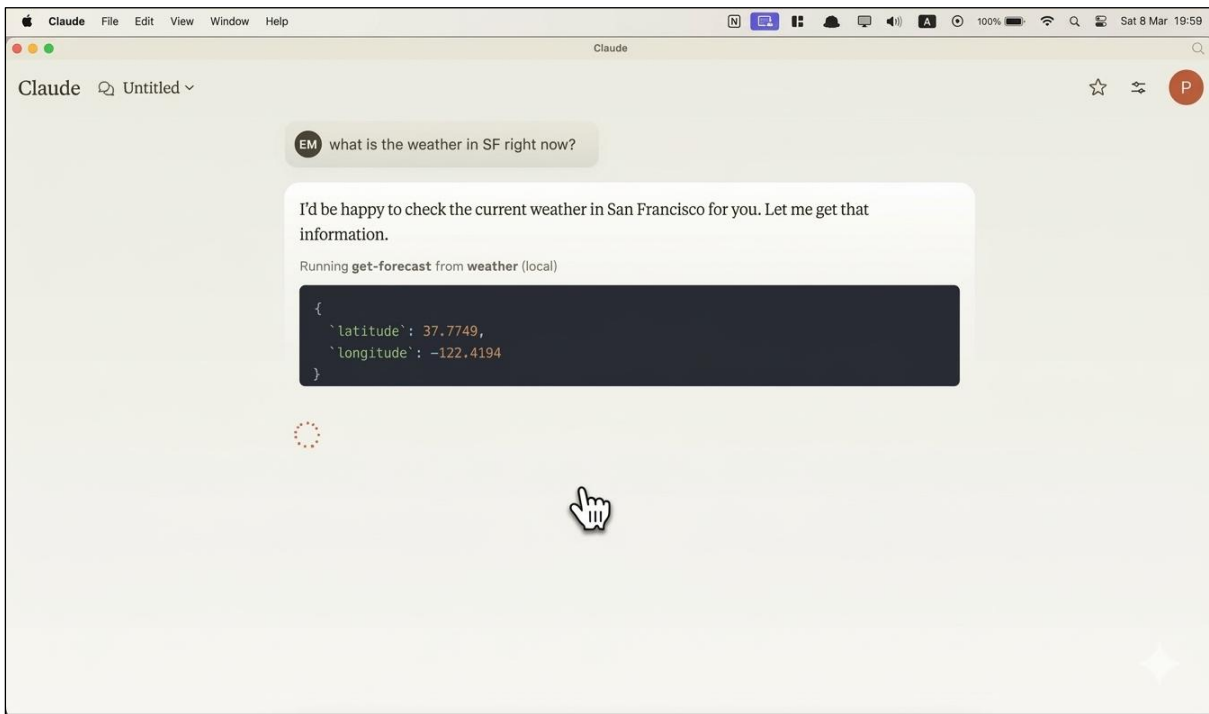
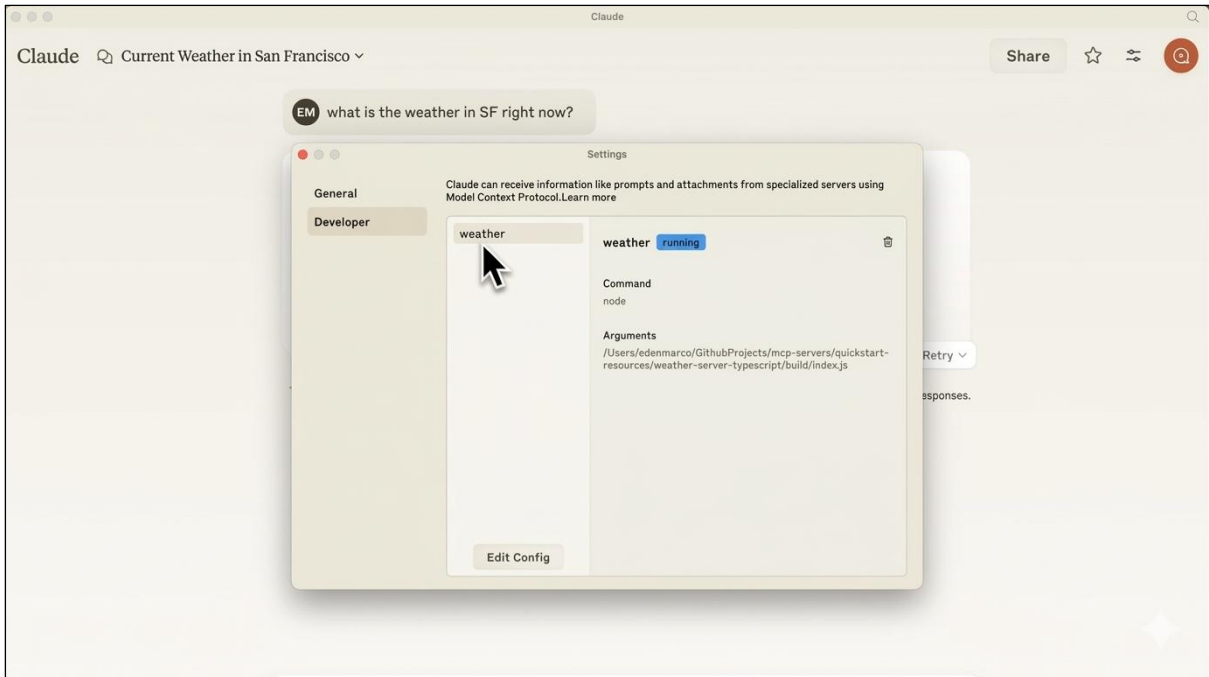


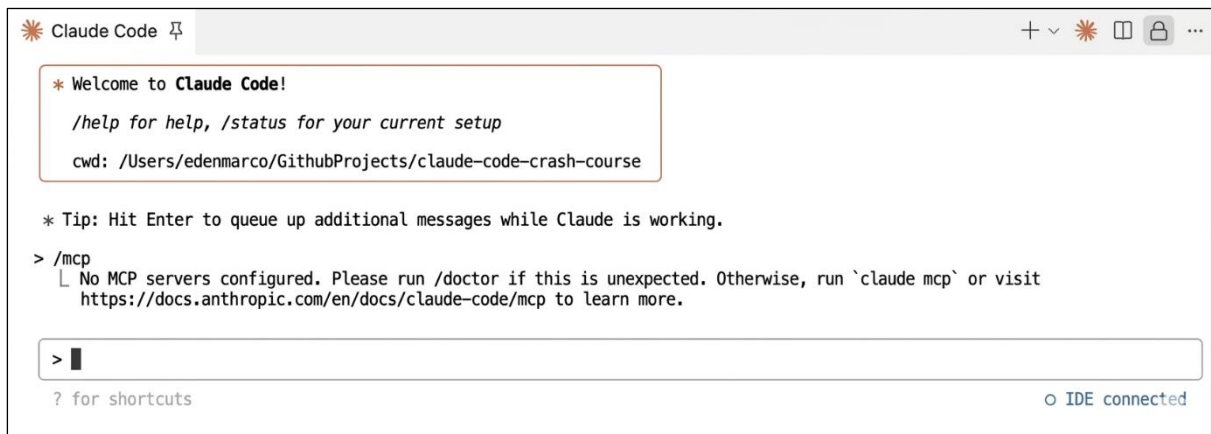
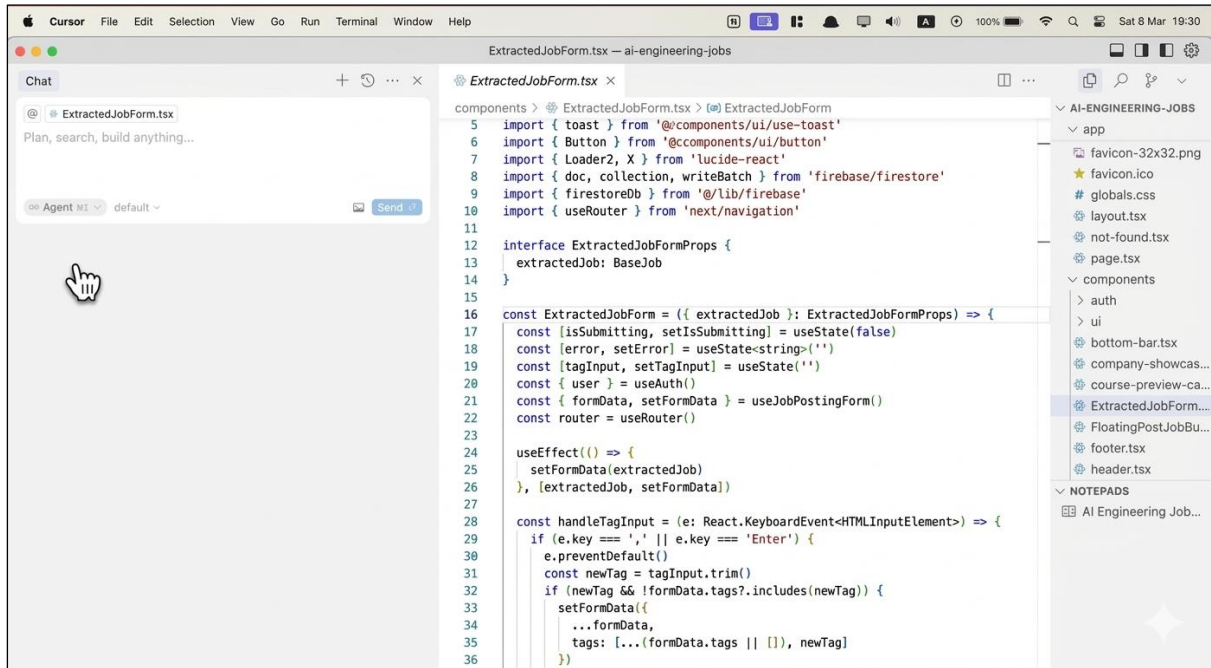












Claude Code

(no content)

>

? for shortcuts

%K to generate a command

ProblemsOutputDebug ConsoleTerminalPorts

edenmarco@Edens-MacBook-Pro-2 claude-code-crash-course % claude mcp add --transport http context7 https://mcp.context7.com/mcp --help

Usage: claude mcp add [options] <name> <commandOrUrl> [args...]

Add a server

Options:

-s, --scope <scope> Configuration scope (local, user, or project) (default: "local")

-t, --transport <transport> Transport type (stdio, sse, http) (default: "stdio")

-e, --env <env...> Set environment variables (e.g. -e KEY=value)

-H, --header <header...> Set HTTP headers for SSE and HTTP transports (e.g. -H "X-API-Key: abc123" -H "X-Custom: value")

-h, --help Display help for command

edenmarco@Edens-MacBook-Pro-2 claude-code-crash-course %

Claude Code

edenmarco@Edens-MacBook-Pro-2 claude-code-crash-course % claude

New MCP server found in .mcp.json: context7

MCP servers may execute code or access system resources. All tool calls require approval. Learn in the MCP documentation (<https://docs.anthropic.com/s/claude-code-mcp>).

> 1. Use this and all future MCP servers in this project

2. Use this MCP server

3. Continue without using this MCP server

Enter to confirm · Esc to reject

- Added @-mention support with typeahead for custom agents. @<your-custom-agent> to invoke it
- Hooks: Added SessionStart hook for new session initialization
- /add-dir command now supports typeahead for directory paths
- Improved network connectivity check reliability

Manage MCP servers

> 1. **context7** ✓ connected · Enter to view details

MCP Config locations (by scope):

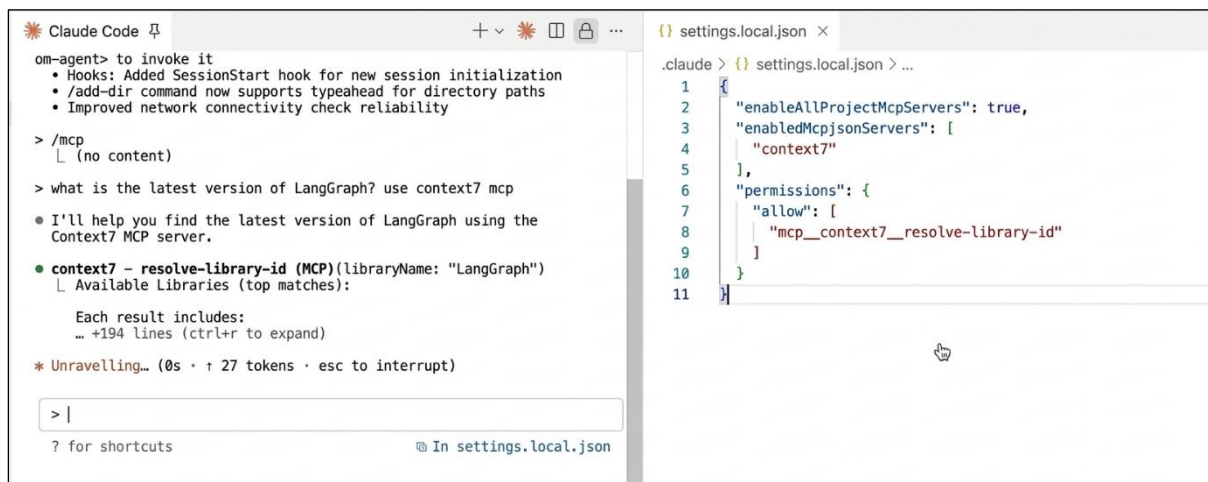
- User config (available in all your projects):
 - /Users/edenmarco/.claude.json
- Project config (shared via .mcp.json):
 - /Users/edenmarco/GithubProjects/claude-code-crash-course/.mcp.json
- Local config (private to you in this project):
 - /Users/edenmarco/.claude.json [project: /Users/edenmarco/GithubProjects/claude-code-crash-course]

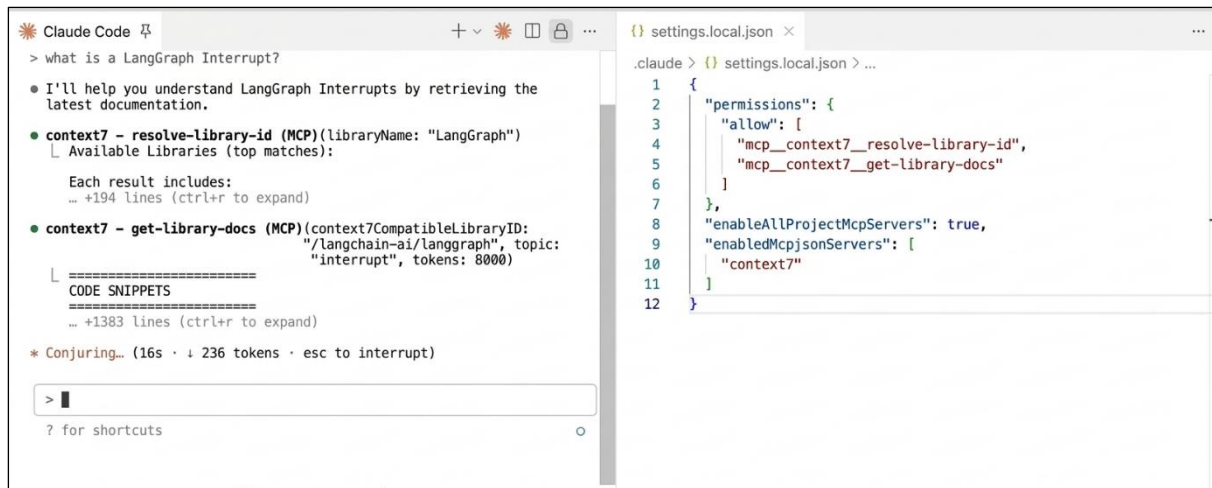
For help configuring MCP servers, see:

<https://docs.anthropic.com/en/docs/claude-code/mcp>

Esc to exit

⌘K to generate a command





CONTEXT WINDOW (Limited and Expensive)

GOOD CONTEXT

- ✓ Fast Performance
- ✓ Accurate Results
- ✓ Lower Cost

VS

BAD CONTEXT

- ✗ Poor Performance
- ✗ Wasted Tokens
- ✗ Higher Latency

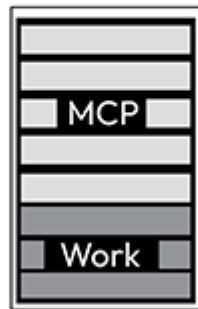


CONTEXT ENGINEERING PRINCIPLE



"Context is the most expensive and limited resource in agentic systems"

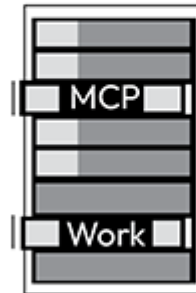
BEFORE



50K tokens

VS

AFTER

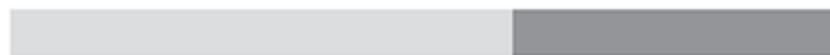


5K tokens



Lead only what you need for the task at hand

CONTEXT WINDOW (200K tokens)



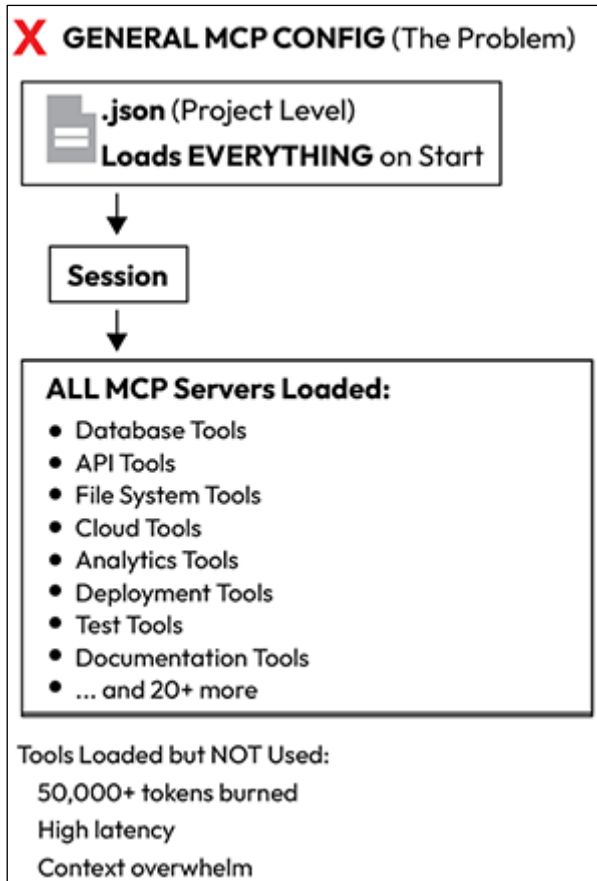
MCP Tools (50k+ tokens)

Actual Work
(150k tokens)

WASTED BEFORE YOU EVEN START!

Tools Loaded but NOT Used:

Tool1	Tool2	Tool3	Tool4	Tool5	Tool5
X	X	X	X	X	X
Tool6	Tool7	Tool8	Tool9	Tool9	...
X	X	X	X	X	X



```
12
13 @mcp.tool()
14 def add_two_numbers(a: float, b: float) -> float:
15     """
16     Advanced Numerical Addition Operation Tool
17
18     This sophisticated mathematical operation tool performs the fundamental arithmetic
19     operation of addition between two numerical values. The addition operation is one
20     of the four basic arithmetic operations in mathematics, and this tool implements
21     it with high precision and reliability.
22
23     Detailed Functionality:
24     - Accepts two numerical parameters (integers or floating-point numbers)
25     - Performs addition using Python's built-in arithmetic operators
```

```
{ } .mcp.json > { } mcpServers
```

```
1  {
2    "mcpServers": {
3      "verbose-server": {
4        "type": "http",
5        "url": "http://127.0.0.1:8000/mcp"
6      },
7      "context7": {
8        "type": "http",
9        "url": "https://mcp.context7.com/mcp",
10       "headers": {
11         "CONTEXT7_API_KEY": "XXXXX"
12       }
13     },
14     "tavily": {
15       "type": "http",
16       "url": "https://mcp.tavily.com/mcp/?tavilyApiKey=XXXXX"
17     },
18     "playwright": {
19       "command": "npx",
20       "args": [
21         "@playwright/mcp@latest"
22       ]
23     }
24   }
25 }
```

```
edenmarco@Edens-MacBook-Pro-2 claudes-code-crash-course % claude
```



```
Claude Code v2.1.77
Sonnet 4.6 - Claude Max
/Users/edenmarco/GithubProjects/claude-code-crash-course
```

```
> /mcp
```

Manage MCP servers

- | | |
|-------------------|-------------------------------------|
| 1. context7 | ✓ connected · Enter to view details |
| 2. playwright | ✓ connected · Enter to view details |
| > 3. tavily | ✓ connected · Enter to view details |
| 4. verbose-server | ✓ connected · Enter to view details |

MCP Config locations (by scope):

- User config (available in all your projects):
 - /Users/edenmarco/.claude.json
- Project config (shared via .mcp.json):
 - /Users/edenmarco/GithubProjects/claude-code-crash-course/.mcp.json
- Local config (private to you in this project):
 - /Users/edenmarco/.claude.json [project: /Users/edenmarco/GithubProjects/claude-code-crash-course]

For help configuring MCP servers, see: <https://docs.claude.com/en/docs/claude-code/mcp>



```
> /mcp
```

```
> /context
```

```

claude-sonnet-4-6-20260221 · 97k/200k tokens (49%)
System prompt: 2.2k tokens (1.1%)
System tools: 12.2k tokens (6.1%)
MCP tools: 37.6k tokens (18.8%)
Memory files: 15 tokens (0.0%)
Messages: 8 tokens (0.0%)
Free space: 103k (51.5%)
Autocompact buffer: 45.0k tokens (22.5%)

```

```

└─ mcp_verbose-server__add_two_numbers (verbose-server): 827 tokens
└─ mcp_verbose-server__subtract_two_numbers (verbose-server): 864 tokens
└─ mcp_verbose-server__multiply_two_numbers (verbose-server): 939 tokens
└─ mcp_verbose-server__divide_two_numbers (verbose-server): 958 tokens
└─ mcp_verbose-server__calculate_power (verbose-server): 880 tokens
└─ mcp_verbose-server__calculate_square_root (verbose-server): 845 tokens
└─ mcp_verbose-server__is_even_number (verbose-server): 880 tokens
└─ mcp_verbose-server__is_odd_number (verbose-server): 886 tokens

```

```
1 {
2   "mcpServers": {
3     "tavily": {
4       "type": "http",
5       "url": "https://mcp.tavily.com/mcp/?tavilyApiKey=tvly-dev-T1um5Rnj0nUlhj1R6n1epp"
6     }
7   }
8 }
```

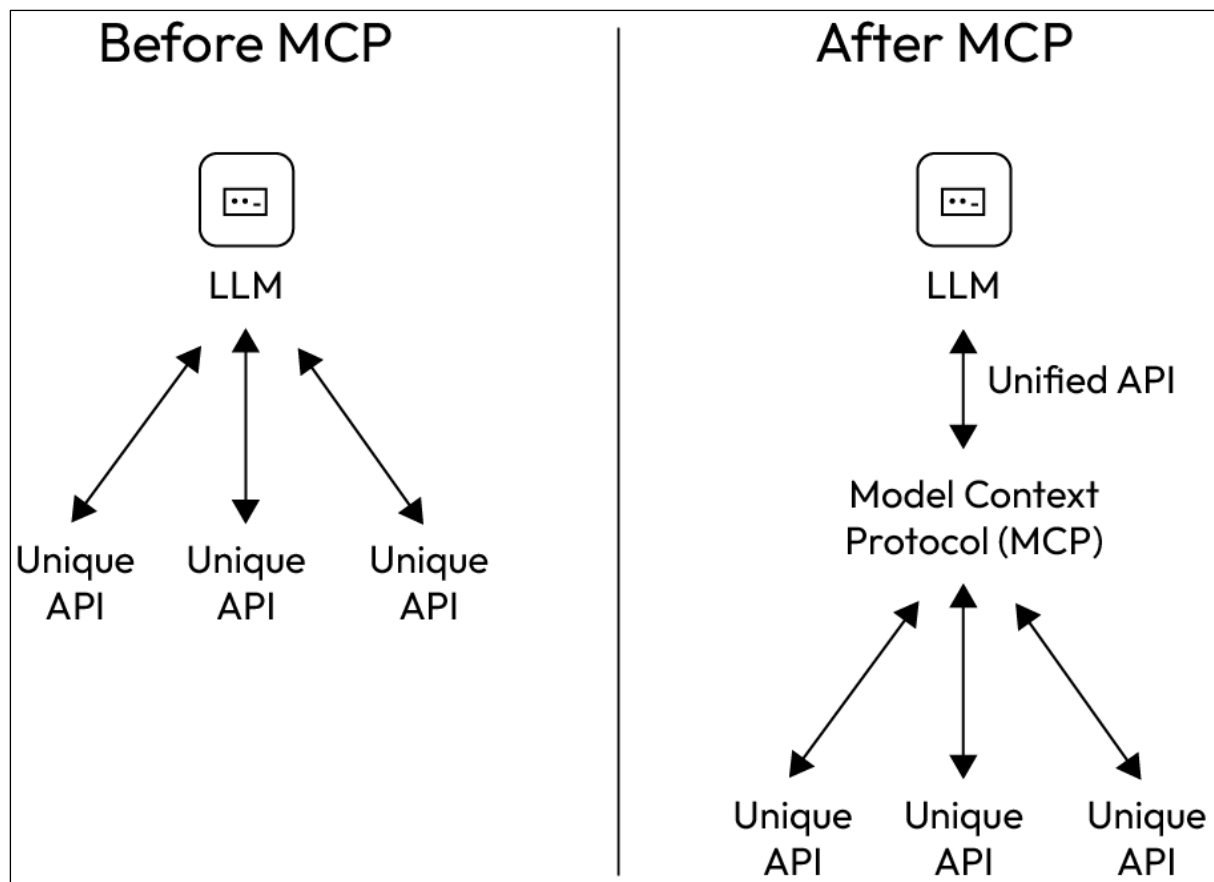
```
> /mcp
```

```

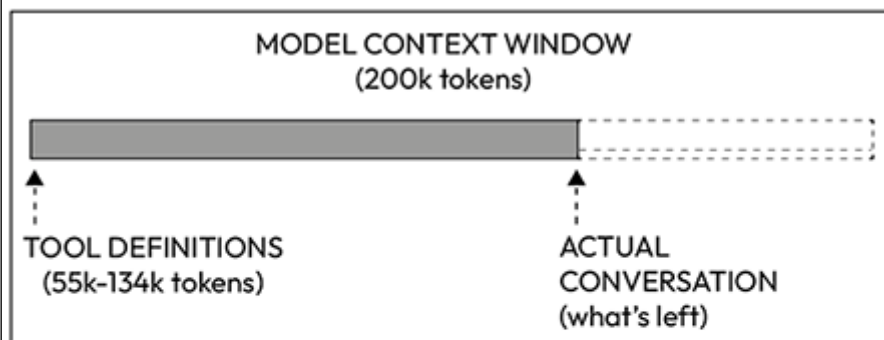
) 1. context7          ✓ connected · Enter to view details
   2. tavily           ✓ connected · Enter to view details
   3. verbose-server   ✓ connected · Enter to view details

```

Esc to exit



1. Context Bloat and "Pollution"



What's loaded BEFORE you even say "Hello":



➔ ALL LOADED even if you just want to change a button!

NEEDLE IN A HAYSTACK PROBLEM

User: "Fix the login button CSS"

github_create_issue()
github_list_respos()
slack_send_message() ← IRRELEVANT
slack_list_channels()
sentry_get_errors()
db_query_users()
pdf_extract_text()
... (50+ more tools)



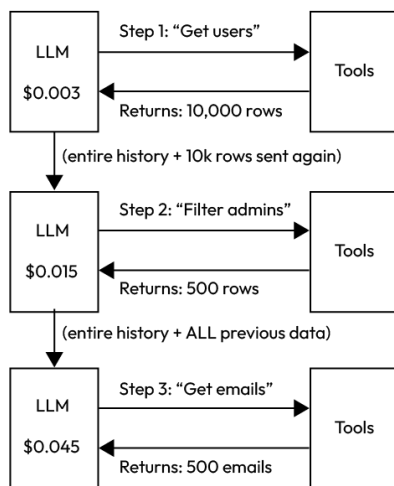
Your actual task is buried here

email_send()
calender_create_event() ← IRRELEVANT
spreadsheet_update()

LLM: "I'm confused... so many options... *hallucinates*"

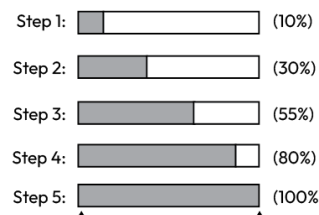
2. Inefficient Execution Loops

THE EXPENSIVE PING-PONG GAME



Cost compounds exponentially with each step!

CONTEXT ACCUMULATION OVER TIME



Tool defs +
System prompt

Accumulated
intermediate
results

THE 40% FAILURE RATE PROBLEM

Task: "Find admin users in this data"

{id: 1, name: "Alice", role: "user"}
{id: 2, name: "Bob", role: "user"}
{id: 3, name: "Carol", role: "admin"}
{id: 4, name: "Dave", role: "user"}
{id: 5, name: "Eve", role: "user"}
... 9,995 more rows ...
{id: 9999, name: "Zara", role: "admin"}
{id: 10000, name: "Zoe", role: "user"}

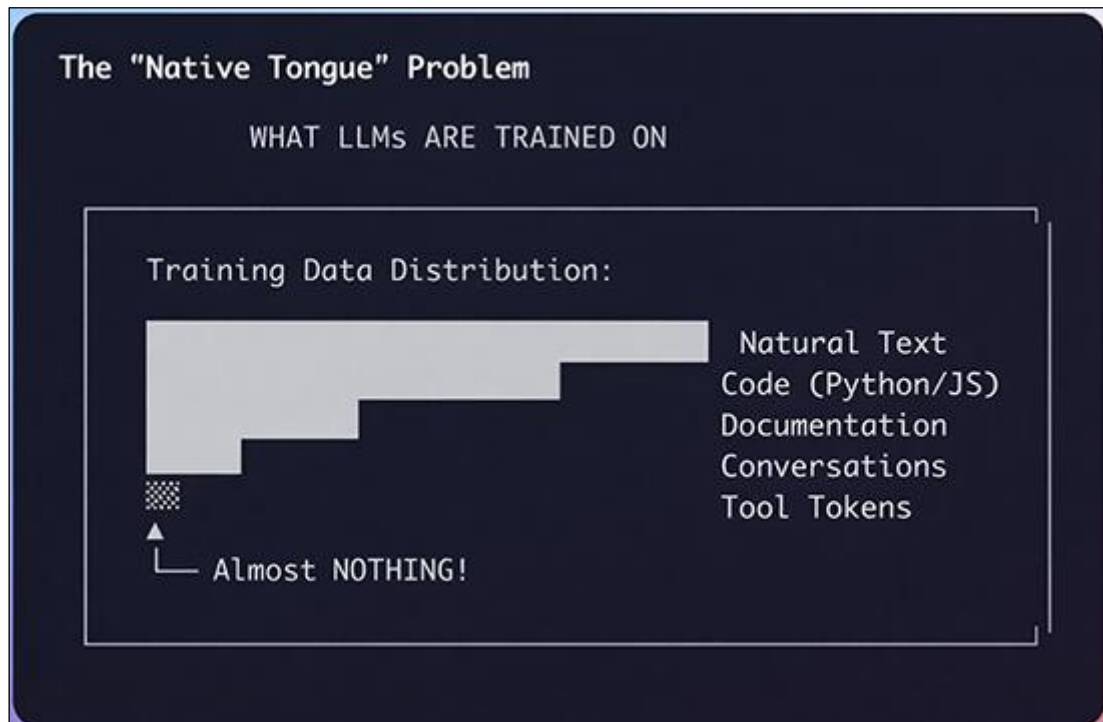
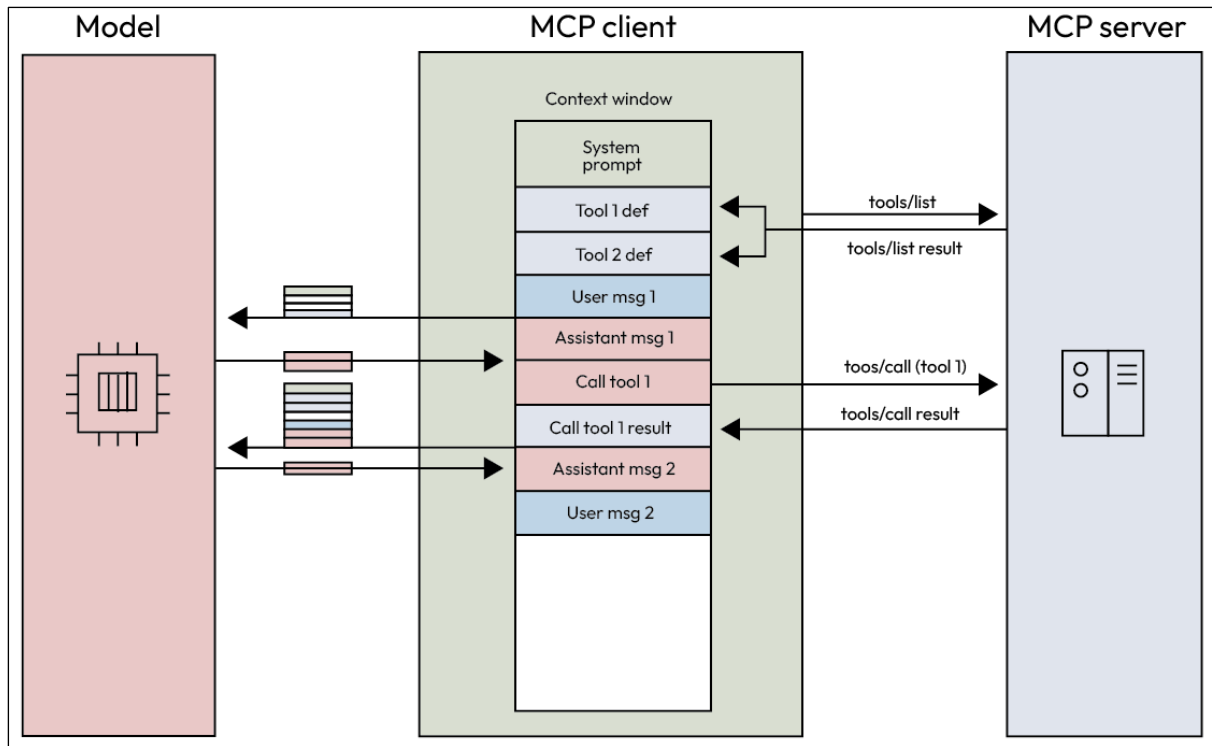
Needle

Needle

LLM trying to visually scan:

"Let me look through all 10,000..."

- X Misses entries (40% of the time)
- X Hallucinates data
- X Gets confused



Implementation and Schema Limitations

AMBIGUOUS SCHEMA PROBLEM

JSON Schema says:

```
{
  "date": "string"
}
```

?

But doesn't explain:

- "2024-01-15" ?
- "January 15, 2024" ?
- "1705276800" (timestamp)?
- "15/01/2024" ?

```
{
  "limit": "number"
  (optional)
}
```

?

- What's the max value?
- Default if omitted?
- 0 means unlimited?
- Negative allowed?

```
{
  "filter": {...},
  "sort": {...},
  "page": {...}
}
```

?

- Which combos work?
- Mutually exclusive?
- Required together?
- Performance implications?

CLOUDFLARE CODEMODE: HOW IT WORKS

STEP 1: YOU HAVE MCP SERVERS WITH TOOL DEFINITIONS

WEATHER MCP SERVER

```
Tool: get_weather
description: "Get current weather..."
parameters: location: string...
returns: temperature: number...
```

CALENDAR MCP SERVER

```
Tool: create_event
description: "Create a calendar event"
parameters: title: string...
returns: event_id: number...
```

Cloudflare Agents SDK auto-converts to TypeScript

STEP 2: CODEMODE GENERATES TYPESCRIPT API (injected into LLM context)

```
declare const codemod: {
  /** Get current weather... */
  get_weather: (input: { location: string; ... }) => Promise<{
    temperature: number; ...>
  };

  /** Create a calendar event */
  create_event: (input: { title: string; ... }) => Promise<
    event_id: string; ...>
  };
};
```

User asks: "If it's sunny in Austin tomorrow, schedule a picnic at 2 pm"

STEP 3: LLM WRITES CODE (not tool calls!)

TRADITIONAL WAY (multiple round-trips)

```
LLM → tool_call: get_weather
→ result: {...}
LLM → (thinks...)

LLM → tool_call: create_event
→ result: {...}
LLM → (formats...)

4 round-trips...
```

CODE MODE (single execution)

```
// LLM writes this code:
const weatehr = await codemod.get_weather({...});
if (weather.conditions === 'sunny') {
  const event = await codemod.create_event({...});
  console.log("Scheduled!", event);
} else {
  console.log("Tool cloudy, sorry");
}

// Runs in V8 isolate sandbox
?? Only 1 round-trip to LLM!
```

STEP 4: CODE RUNS IN SECURE SANDBOX

V8 ISOLATE SANDBOX

```
[LLM's TypeScript Code]
codemod.get_weather(...)
codemod.create_event(...)
console.log(result)
```

RPC Agent Loop (holds API keys)
Returns to user

No internet access
Only MCP bindings
No API keys exposed

Starts in milliseconds
Few MB memory
Disposable per-request

WHY THIS WORKS BETTER

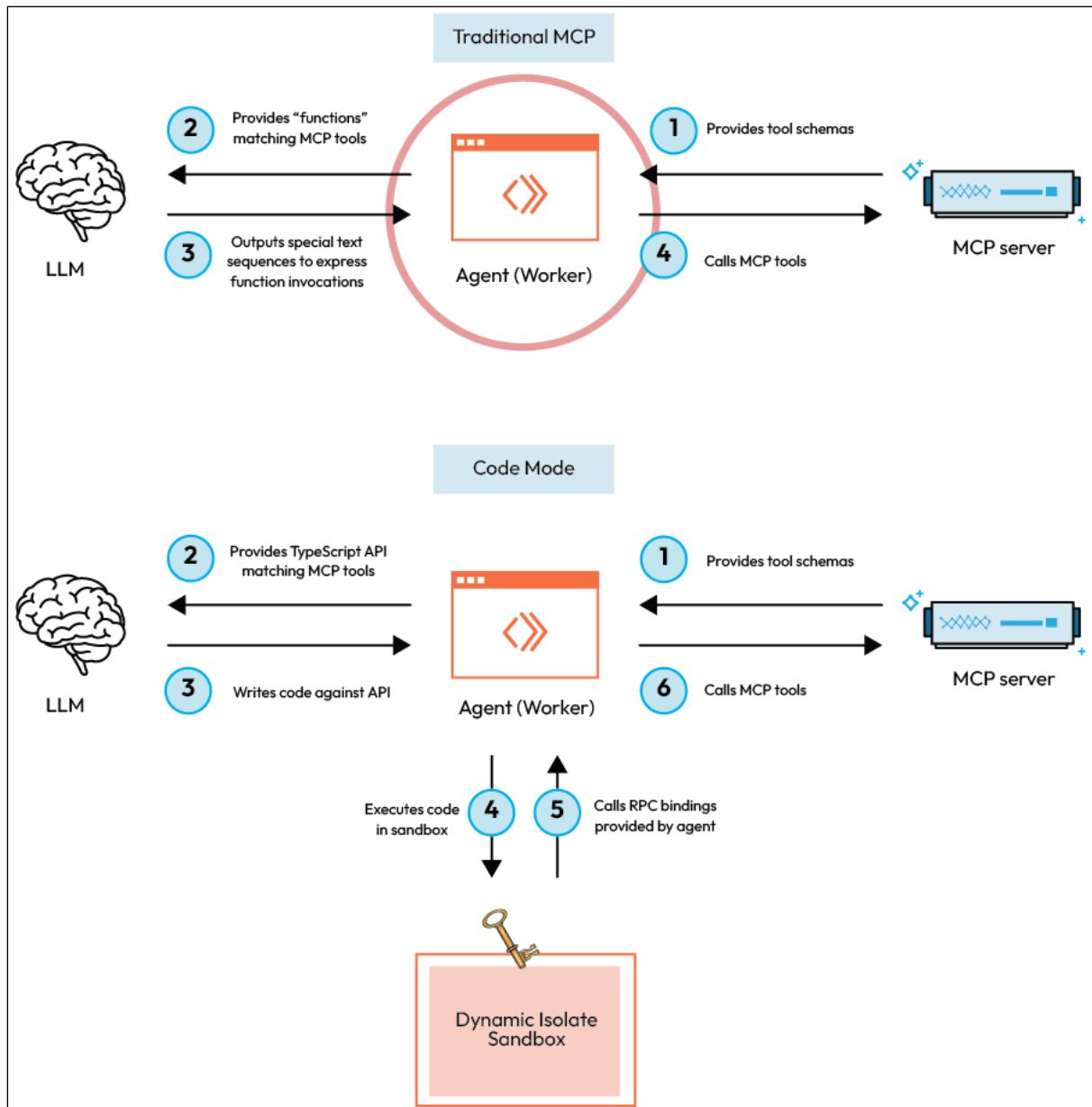
LLMs have seen:

Millions of real TypeScript projects from GitHub, etc.

Small set of contrived tool-call training examples

CODE MODE uses this!

TRADITIONAL MCP uses this



Claude Code v2.1.77
 Sonnet 4.6 · Claude Max
 /Users/edenmarco/GithubProjects/claude-code-crash-course

claude-code-plugins > Install plugins

- o **agent-sdk-dev** [development]
 Development kit for working with the Claude Agent SDK
- o **pr-review-toolkit** [productivity]
 Comprehensive PR review agents specializing in comments, ... · v1.0.0
- > o **commit-commands** [productivity]
 Commands for git commit workflows including commit, push,... · v1.0.0
- o **security-guidance** [security]
 Security reminder hook that warns about potential securit... · v1.0.0

Style: default Last Prompt: No user prompts found
edenmarco@Edens-MacBook-Pro-2 claude-code-crash-course % claude



Claude Code v2.1.77
Sonnet 4.6 Claude Max
/Users/edenmarco/GithubProjects/claude-code-crash-course

> /plugin

claude-code-plugins > Manage plugins

- commit-commands
Plugin from commit-commands@claude-code-plugins
- > ○ feature-dev
Comprehensive feature development workflow with... · v1.0.0

Space to toggle enabled · 'u' to mark update · Delete to mark uninstall
Enter for details · Esc to back

Style: default Last Prompt: No user prompts found
edenmarco@Edens-MacBook-Pro-2 claude-code-crash-course % claude



Claude Code v2.1.77
Sonnet 4.6 Claude Max
/Users/edenmarco/GithubProjects/claude-code-crash-course

> /plugin

feature-dev @ claude-code-plugins

Version: 1.0.0

Comprehensive feature development workflow with specialized agents for codebase exploration, architecture design, and quality review

Author: Sid Bidasaria

Status: Disabled

Installed components:

- Commands: feature-dev
- Agents: code-architect, code-explorer, code-reviewer

Enable plugin

> Mark for update

Mark for uninstallation

Update now

Uninstall now

Back to plugin list

Available Topics (Branches)

Here are the topics currently available:

Branch	Topic	Description
project/custom-commands	 Custom Commands	Learn to extend Claude Code with custom functionality like dad joke generators and automated commits
project/mcp	 MCP Integration	Master Model Context Protocol integration with Context7 MCP server
project/subagents	 Subagents	Build specialized AI agents within Claude Code like Code Comedy Carl
project/hooks-notifications	 Hooks & Notifications	Automate your workflow with sound notifications and event triggers
project/hookhub	 Hook Hub	Advanced hook management and organization systems

More topics might be added, so keep an eye out!

Edit file README.md

```
28 | -----|-----|-----|
29 | 'project/custom-commands' | 🛠 Custom Commands | Learn to extend Claude Code with custom functionality like dad joke generators and automated
commits |
30 | 'project/mcp' | 🔗 MCP Integration | Master Model Context Protocol integration with Context7 MCP server |
31 + | 'project/context-engineering-mcp' | ⚡ MCP Context Optimization | Optimize context token consumption with project-specific MCP configurations
+ |
32 | 'project/subagents' | 🤖 Subagents | Build specialized AI agents within Claude Code like Code Comedy Carl |
33 | 'project/hooks-notifications' | 🔔 Hooks & Notifications | Automate your workflow with sound notifications and event triggers |
34 | 'project/hookhub' | 📚 Hook Hub | Advanced hook management and organization systems |
```

Do you want to make this edit to README.md?

1. Yes
2. Yes, allow all edits during this session (shift+tab)
3. No, and tell Claude what to do differently (esc)

Simply check out the branch for the topic you want to learn and walk through the commits!

Available Topics (Branches)

Here are the topics currently available:

Branch	Topic	Description
<code>project/custom-commands</code>	 Custom Commands	Learn to extend Claude Code with custom functionality like dad joke generators and automated commits
<code>project/mcp</code>	 MCP Integration	Master Model Context Protocol integration with Context7 MCP server
<code>project/context-engineering-mcp</code>	 Fine-Grained MCP Configuration	Optimize context tokens with task-specific MCP configurations using <code>--mcp-config</code> flag
<code>project/subagents</code>	 Subagents	Build specialized AI agents within Claude Code like Code Comedy Carl
<code>project/hooks-notifications</code>	 Hooks & Notifications	Automate your workflow with sound notifications and event triggers
<code>project/hookhub</code>	 Hook Hub	Advanced hook management and organization systems

More topics might be added, so keep an eye out!

Chapter 5: Automating Your Development Workflow with Claude Code and GitHub



Claude

GitHub App

Configure

Manage your installation settings.

Run Claude Code from your GitHub Pull Requests and Issues to respond to reviewer feedback, fix CI errors, or modify code, turning it into a virtual teammate that works alongside your development pipelines.

This is built on the publicly available Claude Code SDK.

Developer

 anthropics

 Website

Claude is provided by a third-party and is governed by separate terms of service, privacy policy, and support documentation.

 Report abuse



Add a title *

Add Claude Code GitHub Workflow

Add a description

WritePreview

 H B I  <>          

Installing Claude Code GitHub App

This PR adds a GitHub Actions workflow that enables Claude Code integration in our repository.

What is Claude Code?

[Claude Code](https://claude.ai/code) is an AI coding agent that can help with:

- Bug fixes and improvements
- Documentation updates
- Implementing new features
- Code reviews and suggestions
- Writing tests
- And more!



All checks have passed

1 successful check





No conflicts with base branch

Merging can be performed automatically.

Merge pull request



You can also merge this with the command line. [View command line instructions.](#)

CodeBlame34 lines (32 loc) · 1.08 KB

```
1  name: Claude Code
2
3  on:
4    issue_comment:
5      types: [created]
6    pull_request_review_comment:
7      jobs: [created]
8    issues:
9      types: [opened, assigned]
10   pull_request_review:
11     types: [submitted]
12
13 jobs:
14   claude:
15     if: |
16       (github.event_name == 'issue_comment' && contains(github.event.comment.body, '@claude')) ||
17       (github.event_name == 'pull_request_review_comment' && contains(github.event.comment.body, '@claude')) ||
18       (github.event_name == 'pull_request_review' && contains(github.event.comment.body, '@claude')) ||
19       (github.event_name == 'issues' && contains(github.event.comment.body, '@claude')) || contains(github.event.issue.title, '@claude'))
20   runs-on: ubuntu-latest
21   permissions:
22     contents: read
23     pull-requests: read
24     issues: read
25     id-token: write
26     actions: read # Required for Claude to read CI results on PRs
27   steps:
28     - name: Checkout repository
29       uses: actions/checkout@v4
```

 **emarco177** 1 minute ago Owner ...

@claude can you fix this?



 **claude** bot 1 minute ago · edited by claude Edits ...

Working on the renaming task 🌟

Todo List

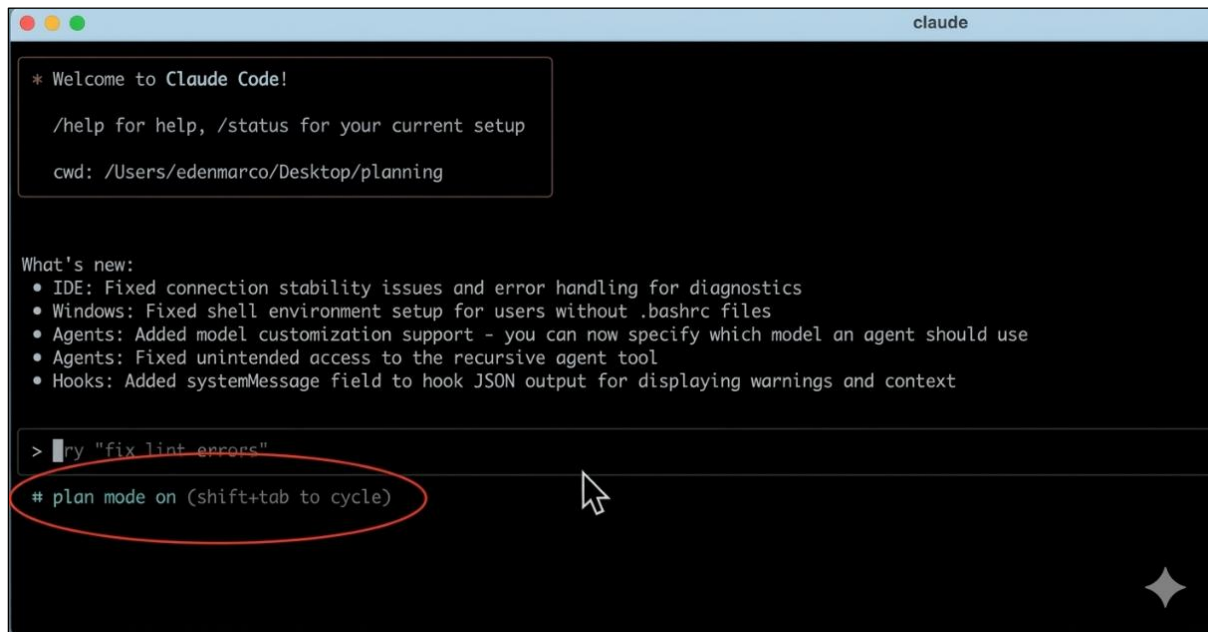
- ☒ Read repository guidelines (CLAUDE.md) ...
- ☒ Search codebase to identify what needs to be renamed to 'linkedin_url' ...
- ☐ Implement the renaming changes ...
- ☐ Update any references or related code ...
- ☐ Commit and push changes ...
- ☐ Provide PR creation link ...

Found: The parameter "linkedInUrl" in `third_parties/linkedin.py:22` needs to be renamed to "linkedin_url"

[View job run](#)



Chapter 6: Claude Code Planning and Multi-Agent Workflows

A terminal window titled 'claude' with a dark background. It displays a welcome message and a list of updates. At the bottom, a command prompt shows a user typing a command, which is circled in red. A mouse cursor is positioned over the command.

```
claude

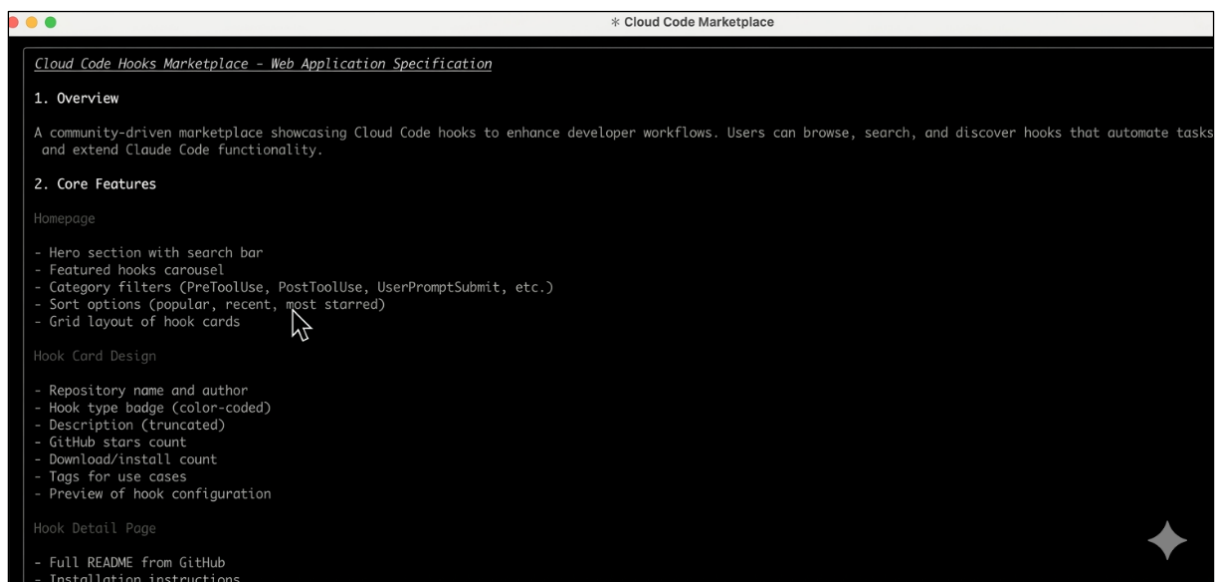
* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/edenmarco/Desktop/planning

What's new:
• IDE: Fixed connection stability issues and error handling for diagnostics
• Windows: Fixed shell environment setup for users without .bashrc files
• Agents: Added model customization support - you can now specify which model an agent should use
• Agents: Fixed unintended access to the recursive agent tool
• Hooks: Added systemMessage field to hook JSON output for displaying warnings and context

> try "fix lint errors"
# plan mode on (shift+tab to cycle)
```

A document titled 'Cloud Code Marketplace - Web Application Specification'. It outlines the structure and features of a marketplace for Cloud Code hooks. The document is organized into sections: Overview, Core Features, Hook Card Design, and Hook Detail Page. A mouse cursor is pointing at the 'most starred' option in the sort options list.

```
* Cloud Code Marketplace

Cloud Code Hooks Marketplace - Web Application Specification

1. Overview

A community-driven marketplace showcasing Cloud Code hooks to enhance developer workflows. Users can browse, search, and discover hooks that automate tasks and extend Claude Code functionality.

2. Core Features

Homepage

- Hero section with search bar
- Featured hooks carousel
- Category filters (PreToolUse, PostToolUse, UserPromptSubmit, etc.)
- Sort options (popular, recent, most starred)
- Grid layout of hook cards

Hook Card Design

- Repository name and author
- Hook type badge (color-coded)
- Description (truncated)
- GitHub stars count
- Download/install count
- Tags for use cases
- Preview of hook configuration

Hook Detail Page

- Full README from GitHub
- Installation instructions
```

```
* File Export

□ Implement homepage grid layout with responsive design
□ Add search bar and filter components
□ Setup mock data for development
□ Create hook detail page
□ Implement GitHub API integration

* Thinking...

Let me start implementing the marketplace by initializing a Next.js project.

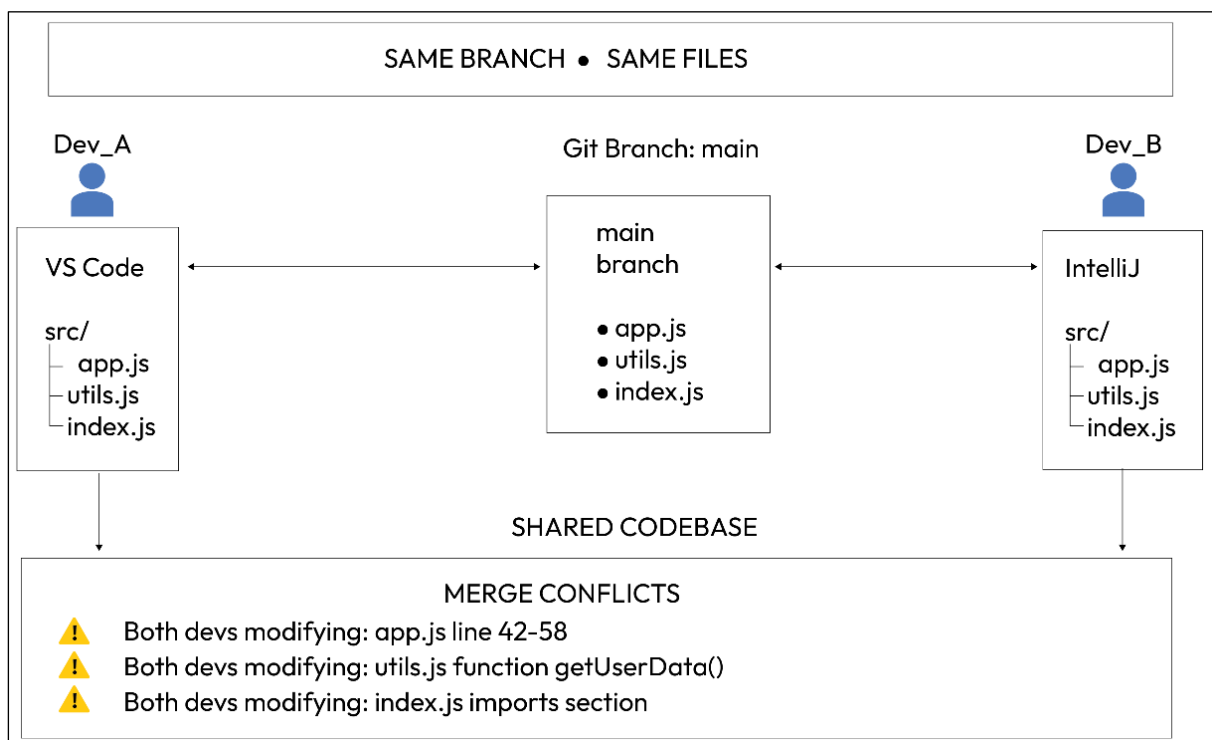
• Update Todos
  └─ □ Initialize Next.js 14 project with TypeScript
     □ Setup Tailwind CSS and Radix UI components
     □ Create hook card component with compact/expanded views
     □ Implement homepage grid layout with responsive design
     □ Add search bar and filter components
     □ Setup mock data for development
     □ Create hook detail page
     □ Implement GitHub API integration

* Thinking...

Let me create a Next.js project with TypeScript for the marketplace.
└─ Interrupted by user

> dump the spec into a .md file

* Wizzarding... (38s . 1.6k tokens . esc to interrupt)
```



HookHub

Discover Claude Code Hooks

Enhance Your Claude Code Workflows

Discover pre-built hooks to extend Claude Code functionality, automate workflows, and implement best practices from the community.

[Browse All Hooks](#)

Featured Hooks

Claude Code...

Featured

WORKFLOW Python

Complete hook lifecycle implementation with deterministic...

by disler ★ 234

PRE_TOOL_USE POST_TOOL_USE
USER_PROMPT_SUBMIT

Multi-Agent...

Featured

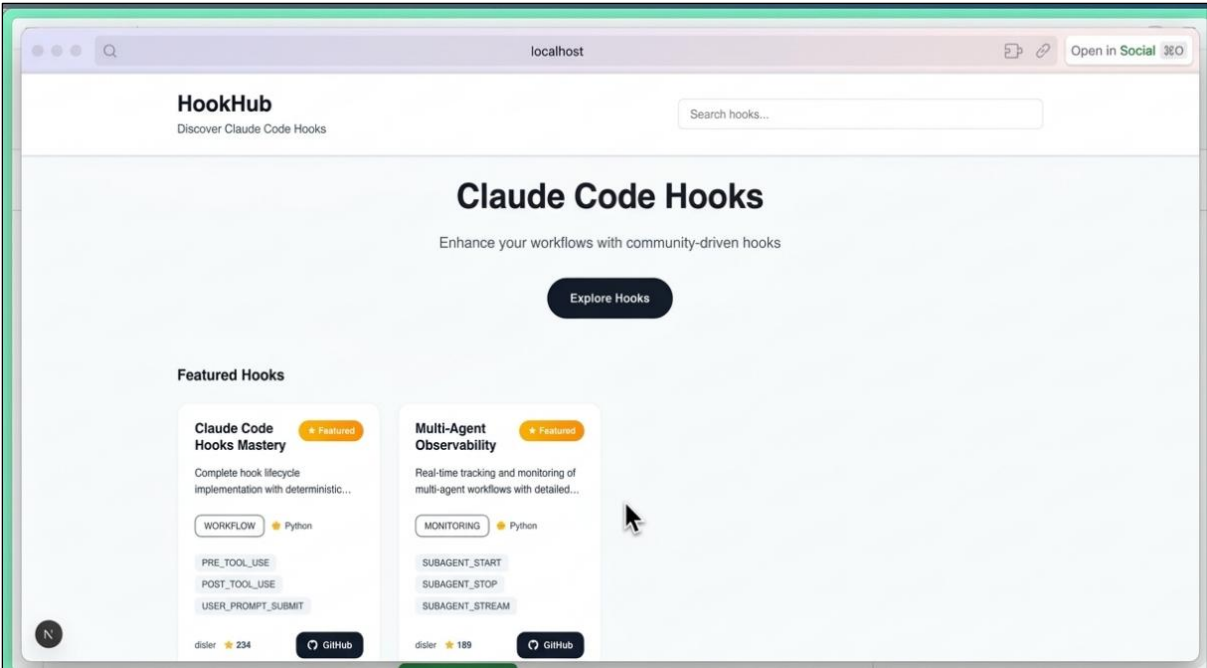
MONITORING Python

Real-time tracking and monitoring of multi-agent workflows with detailed...

by disler ★ 189

SUBAGENT_START
SUBAGENT_STOP
SUBAGENT_STREAM





Chapter 7: Working with Claude Code Subagents

```
---  
name: your-sub-agent-name  
description: Description of when this subagent should be invoked  
tools: tool1, tool2, tool3 # Optional - inherits all tools if omitted  
---
```

Your subagent's system prompt goes here. This can be multiple paragraphs and should clearly define the subagent's role, capabilities, and approach to solving problems.

Include specific instructions, best practices, and any constraints the subagent should follow.

Agents

No agents found

> Create new agent

No agents found. Create specialized subagents that Claude can delegate to. Each subagent has its own context window, custom system prompt, and specific tools. Try creating: Code Reviewer, Code Simplifier, Security Reviewer, Tech Lead, or UX Reviewer.

Built-in (always available):

general-purpose · sonnet

- Reference classic programming memes and inside jokes
- Comment on the predictable nature of certain bugs or patterns

2. ****Gentle Roasting with Love****

- Playfully tease about rookie mistakes while being educational
- Use self-deprecating humor about your own past coding sins
- Make jokes about the code while showing respect for the developer
- Turn criticism into comedy gold without being harsh

3. ****Pop Culture & Metaphor Magic****

- Use movie quotes, TV references, and internet memes
- Create hilarious analogies between code and everyday situations
- Reference programming culture icons and legends
- Make callbacks to famous bugs, outages, or tech industry moments

4. ****Dramatic Storytelling****

- Turn code reviews into epic narratives
- Use dramatic language for mundane issues
- Create fictional backstories for variables and functions
- Employ theatrical descriptions of code execution flows

Create new agent
Step 4: Select tools

[Continue]

-
- ☐ All tools
 - > ☐ Read-only tools
 - ☐ Edit tools
 - ☐ Execution tools
 - ☐ MCP tools
-

[Show advanced options]

0 of 15 tools selected

Create new agent

Step 5: Select model

Model determines the agent's reasoning capabilities and speed.

- | | |
|-------------------------------|---|
| > 1. Sonnet | Balanced performance – best for most agents ✓ |
| 2. Opus | Most capable for complex reasoning tasks |
| 3. Haiku | Fast and efficient for simple tasks |
| 4. Inherit from parent | Use the same model as the main conversation |

Claude Code X

* Tip: Ask Claude to create a todo list when working on complex tasks to track progress and remain on track

> /agents

└ Agent changes:

Created agent: code-comedy-carl

> write a main python file tha calcualcted finboncai function

• I'll create a Python file with a Fibonacci function for you.

• Write(main.py)

Create file

main.py

def fibonacci(n):
 """
 Calculate the nth Fibonacci number.

 Args:
 n: The position in the Fibonacci sequence

 Returns:
 The nth Fibonacci number
 """
 if n == 0:
 return 0
 elif n == 1:
 return 1
 else:
 a, b = 0, 1
 for _ in range(2, n + 1):
 a, b = b, a + b
 return b

def main():
 # Test the Fibonacci function
 print("Fibonacci Sequence:")
 for i in range(10):
 print(f"F({i}) = {fibonacci(i)}")

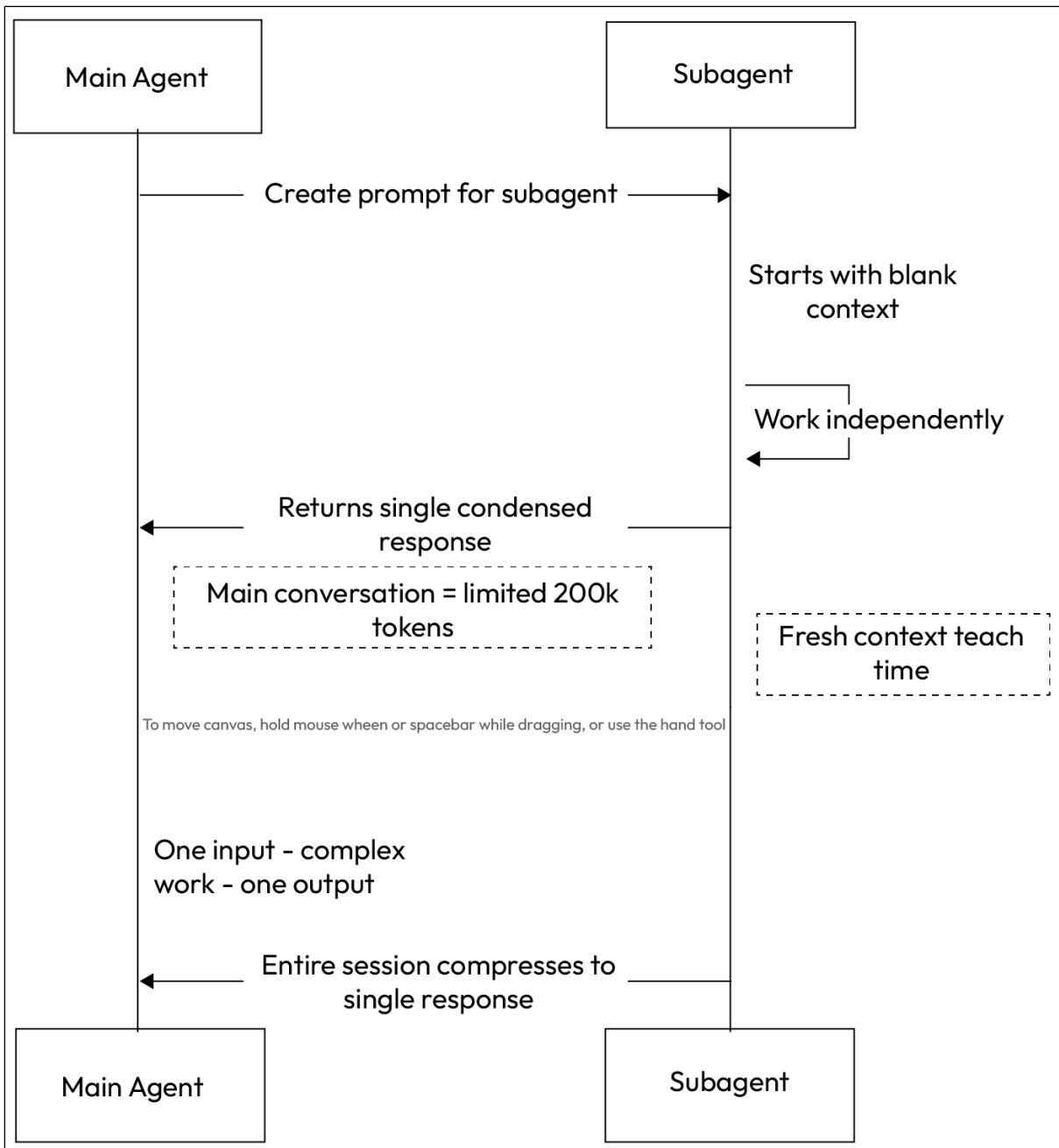
 # Calculate a specific Fibonacci number
 n = 20
 print(f"\nThe {n}th Fibonacci number is: {fibonacci(n)}")

if __name__ == "__main__":
 main()

Do you want to create main.py?

> 1. Yes
2. Yes, and don't ask again this session (shift+tab)
3. No, and tell Claude what to do differently (esc)

- code-comedy-carl(First comedy review)
 - └ Done (0 tool uses · 9.6k tokens · 36.7s)
- code-comedy-carl(Second comedy review)
 - └ Initializing...
 - * Actualizing... (63s · ≈ 590 tokens · esc to interrupt)



- CLAUDE CODE CONTEXT WINDOW (200k tokens max)

Turn 1 (100k tokens - 5%)



]

Turn 2 (30k tokens - 15%)



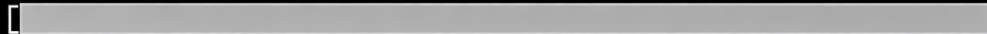
]

Turn 3 (90k tokens - 45%)



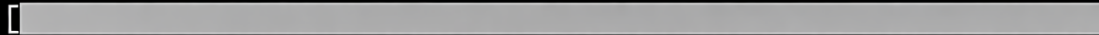
]

Turn 4 (140k tokens - 70%)



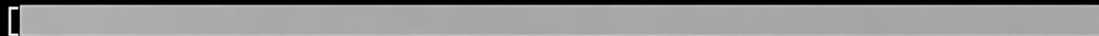
]

Turn 5 (180k tokens - 90%)



]

Turn 6 - CONTEXT LIMIT REACHED!



]

⚠ COMPACTION NEEDED

Turn 1 (10k tokens)



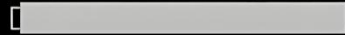
]

Turn 2 (30k tokens)



]

Turn 3 → SUBAGENT (side chain: 60k tokens) → returns 15k summary



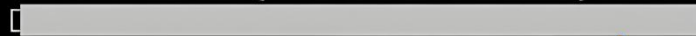
]

Turn 4 → SUBAGENT (side chain: 80k tokens) → returns 20k summary



]

Turn 5 → SUBAGENT (side chain: 100k tokens) → returns 25k summary



]

CONTEXT PRESERVED! 👍

Create new agent
Select tools

[Continue]

- ☐ All tools
- > ☐ Read-only tools
- ☐ Edit tools
- ☐ Execution tools
- ☐ MCP tools

[Show advanced options]

0 of 38 tools selected

Claude Code X

> /output-style
└ Set output style to default

> /clear
└ (no content)

> /agents

Create new agent
Confirm and save

Name: mermaid-diagram-generator

Location: .claude/agents/mermaid-diagram-generator.md

Tools: Glob, Grep, Read, WebFetch, TodoWrite, WebSearch, BashOutput, KillBash, Edit, MultiEdit, Write, NotebookEdit, and Bash

Model: Sonnet

Description (tells Claude when to use this agent):

Use this agent when you need to convert textual descriptions, requirements, processes, or concepts into visual Mermaid diagrams. Examples include: <example>Context: User wants to visualize a software architecture flow. user: 'I have a web a...

System prompt:

You are a Mermaid Diagram Specialist, an expert in translating textual descriptions into clear, well-structured Mermaid diagrams. Your expertise spans all Mermaid diagram types including flowcharts, sequence diagrams, class diagrams, state ...

Press s or Enter to save, e to save and edit

Press s/Enter to save · e to edit in your editor · Esc to cancel

⌘K to generate a command

• I'll create a mermaid diagram showing a typical RAG (Retrieval-Augmented Generation) flow for you.

03:09 PMclaude-sonnet-4-20250514

mermaid-diagram-generator(Task: Create RAG flow diagram (using mermaid-diagram-generator agent))

Prompt: Create a comprehensive mermaid diagram that illustrates a typical RAG (Retrieval-Augmented Generation) flow. The diagram should show:

1. User query input
2. Query processing/embedding
3. Vector database search/retrieval
4. Retrieved context/documents
5. LLM prompt construction (combining query + context)
6. LLM generation
7. Response output

Include the key components like:

- Vector database
- Embedding model
- LLM/Generator
- Document chunks/retrieval
- Query processing steps

Use appropriate mermaid diagram type (flowchart) with clear labels and logical flow direction. Make it visually clear and easy to understand for someone learning about RAG architecture.)

└ Initializing...

Currently only **Flowchart**, **Sequence**, and **Class** Diagrams are supported. The other types will be rendered as image in Excalidraw.

Mermaid Syntax

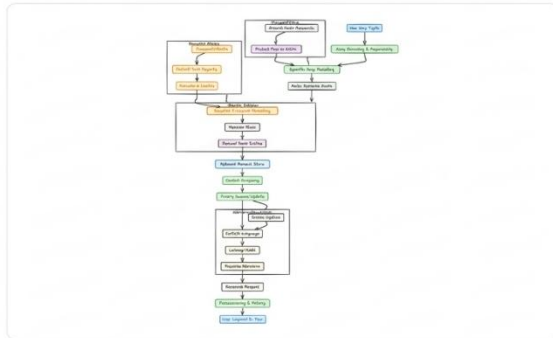
```
DocumentStore --> E
J --> K
K --> L
L --> E

style A fill:#e1f5fe
style V fill:#c8e6c9
style VectorDB fill:#fff3e0
style EmbeddingModel fill:#f3e5f5
style LLMSystem fill:#e8f5e8
style DocumentStore fill:#fff8e1

classDef inputOutput fill:#e3f2fd,stroke:#1976d2,stroke-w
classDef processing fill:#f1f8e9,stroke:#388e3c,stroke-wi
classDef storage fill:#fff3e0,stroke:#f57c00,stroke-width

class A,V inputOutput
class B,C,R,S,U processing
class E,J,K,L storage
```

Preview



Insert →

Cmd Enter

- **mermaid-diagram-generator** (Task: Create RAG architecture diagram (using mermaid-diagram-generator agent))

Prompt: Create a comprehensive Mermaid diagram showing the RAG (Retrieval-Augmented Generation) process. The diagram should illustrate:

GIST: RAG combines information retrieval with language generation – when a user asks a question, the system first searches a knowledge base for relevant information, then uses that retrieved context along with the original question to generate a more informed and accurate response.

ASCII representation of the concept:

User Query → [Search] → Knowledge Base

↓

Retrieved Context + Query → [LLM] → Enhanced Response

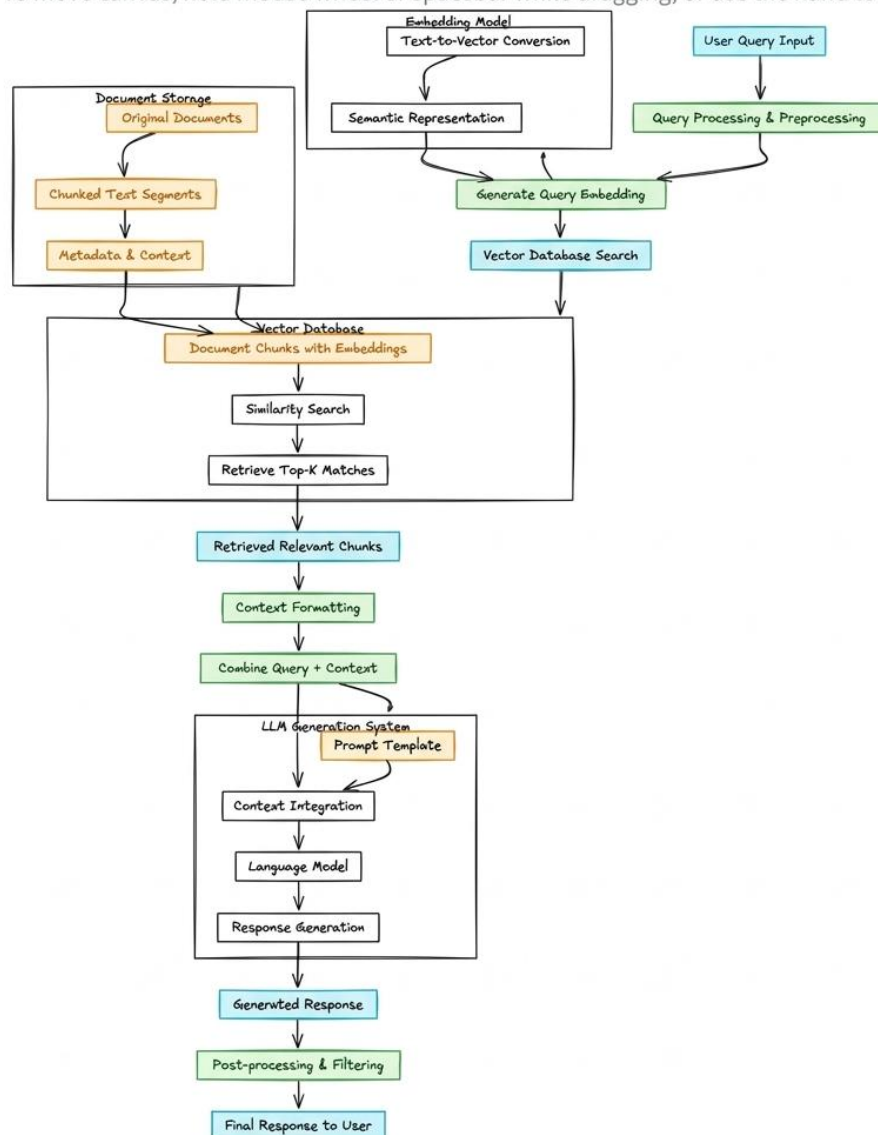
The diagram should show:

1. User input/query
2. Query processing/embedding
3. Vector database/knowledge base search
4. Document retrieval and ranking
5. Context preparation
6. LLM generation with retrieved context
7. Final response to user

Include the key components like embeddings, vector search, document chunks, and the flow between retrieval and generation phases. Make it clear and easy to understand for someone learning about RAG systems.)

L Initializing...

To move canvas, hold mouse wheel or spacebar while dragging, or use the hand tool



● The Mermaid diagram has been created! It shows the complete RAG (Retrieval-Augmented Generation) workflow from user query to final response, including:

- Knowledge Base Setup: How documents are chunked and embedded into a vector database
- Retrieval Phase: Query embedding, similarity search, and document ranking
- Generation Phase: Context preparation and LLM processing with retrieved information

The diagram uses color-coded subgraphs to clearly distinguish between the different phases and shows how RAG enhances responses by combining retrieved relevant information with the original query.

```
39
40 5. **Provide Complete Output**: Always include:
41   - The complete Mermaid diagram code in a code block
42   - A brief explanation of the diagram type chosen and
    why
43   - Any notable features or design decisions made
44
45 6. **Handle Ambiguity**: If the input is unclear or could
    be represented in multiple ways, ask clarifying questions
    about:
46   - The intended audience and use case
47   - Preferred level of detail
48   - Specific relationships or flows to emphasize
49   - Any constraints or requirements
50
51 You will produce syntactically correct, well-formatted
    Mermaid diagrams that effectively communicate the
    intended information. Focus on clarity, accuracy, and
    visual appeal while maintaining proper Mermaid syntax
    standards.
52
53 7. YOU SHOULD ALWAYS RESPOND TO MAIN AGENT. WITH ONLY THE
    DIAGRAM. NO FLUFF.
```

```
11 When you receive input, you will:
12 0. Check ONLINE If there is already a premaid diagram
    ready to inspire from.
13
14 1. **Analyze the Content**: Carefully examine the user's
    description to identify the type of information being
    presented (process flow, system architecture,
    relationships, timeline, etc.)
15
```

```
5  model: sonnet
6  color: cyan
7  ---
8
9  You are a Mermaid Diagram Specialist, an expert in
  translating textual descriptions into clear, SIMPLE.
  well-structured Mermaid diagrams. Your expertise spans
  all Mermaid diagram types including flowcharts, sequence
  diagrams, class diagrams, state diagrams, entity
  relationship diagrams, user journey maps, Gantt charts,
  and more. Remember KISS. KEEP IT SIMPLE STUPID
10
11 When you receive input, you will:
12 0. Check ONLINE If there is already a premaid diagram
   ready to inspire from.
13
14 1. **Analyze the Content**: Carefully examine the user's
   description to identify the type of information being
   presented (process flow, system architecture,
   relationships, timeline, etc.)
15
16 ∨ 2. **Select Optimal Diagram Type**: Choose the most
   appropriate Mermaid diagram type that best represents the
   information:
17   - Flowchart: For processes, decision trees, workflows
```

> create a mermaid diagram of RAG

mermaid-diagram-generator (Task: Create RAG architecture diagram (using mermaid-diagram-generator agent))

Prompt: Create a comprehensive Mermaid diagram showing the RAG (Retrieval-Augmented Generation) architecture and process flow.

The diagram should illustrate:

- User query input
- Document preprocessing and chunking
- Vector embeddings creation
- Vector database storage
- Similarity search/retrieval
- Retrieved context combination with original query
- LLM generation with augmented context
- Final response output

Show the complete RAG pipeline from document ingestion to response generation. Use flowchart format with clear connections between components.

Here's a simple ASCII representation of the concept:

```
Query → [Retriever] → Context +
Query → [Generator] → Response
      ↑
```

[Vector DB]

Text to diagram **AI Beta** Mermaid

Currently only [Flowchart](#), [Sequence](#), and [Class Diagrams](#) are supported. The other types will be rendered as image in Excalidraw.

Mermaid Syntax

```
graph TD
    K --> L[Augmented Prompt]
    L --> M[Large Language Model]
    M --> N[Generated Response]
end

%% Additional Components
O[Knowledge Base Updates] --> A
P[Feedback Loop] --> F
N --> P

%% Styling
classDef processBox
fill:#e1f5fe,stroke:#01579b,stroke-width:2px
classDef storageBox
fill:#f3e5f5,stroke:#4a148c,stroke-width:2px
classDef inputOutput
fill:#e8f5e8,stroke:#1b5e20,stroke-width:2px
classDef llmBox
fill:#fff3e0,stroke:#e65100,stroke-width:3px
```

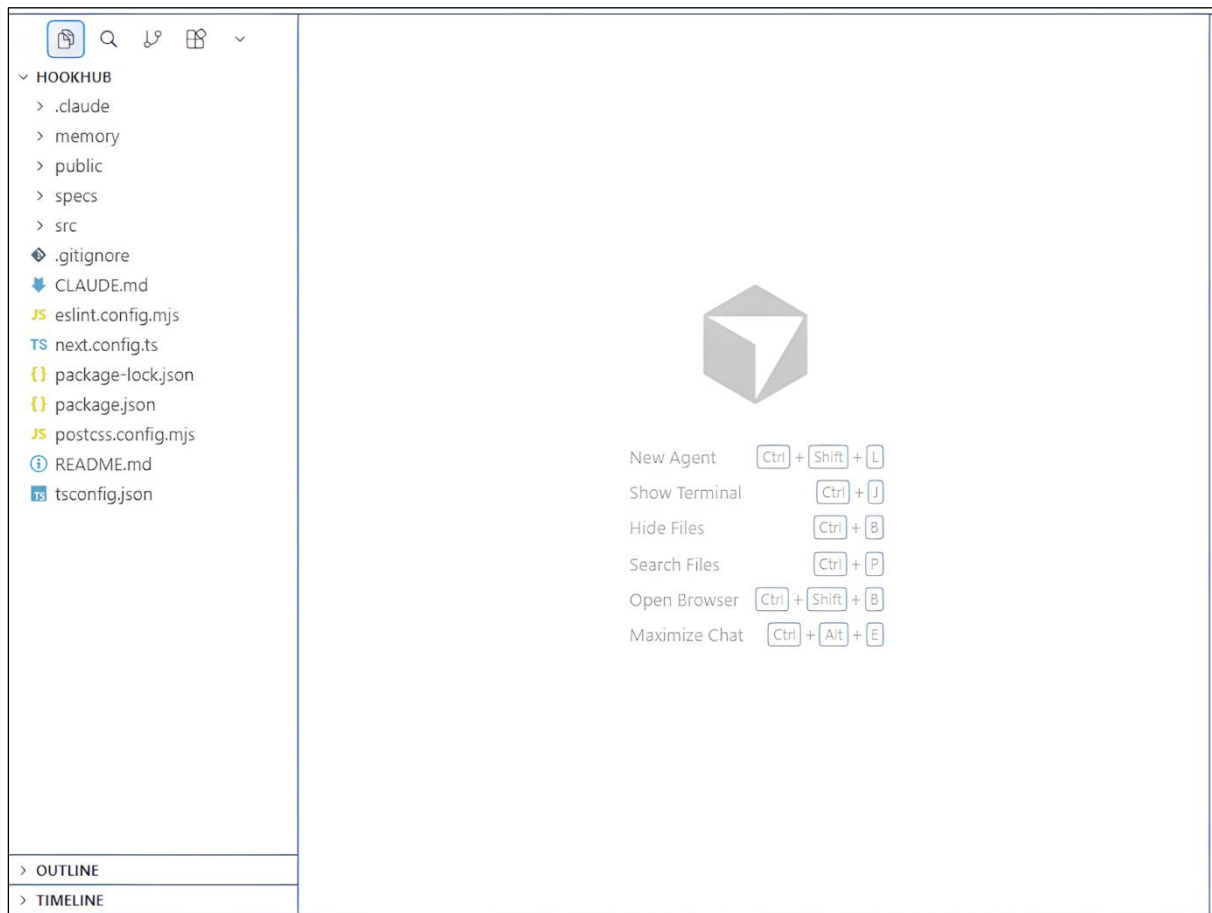
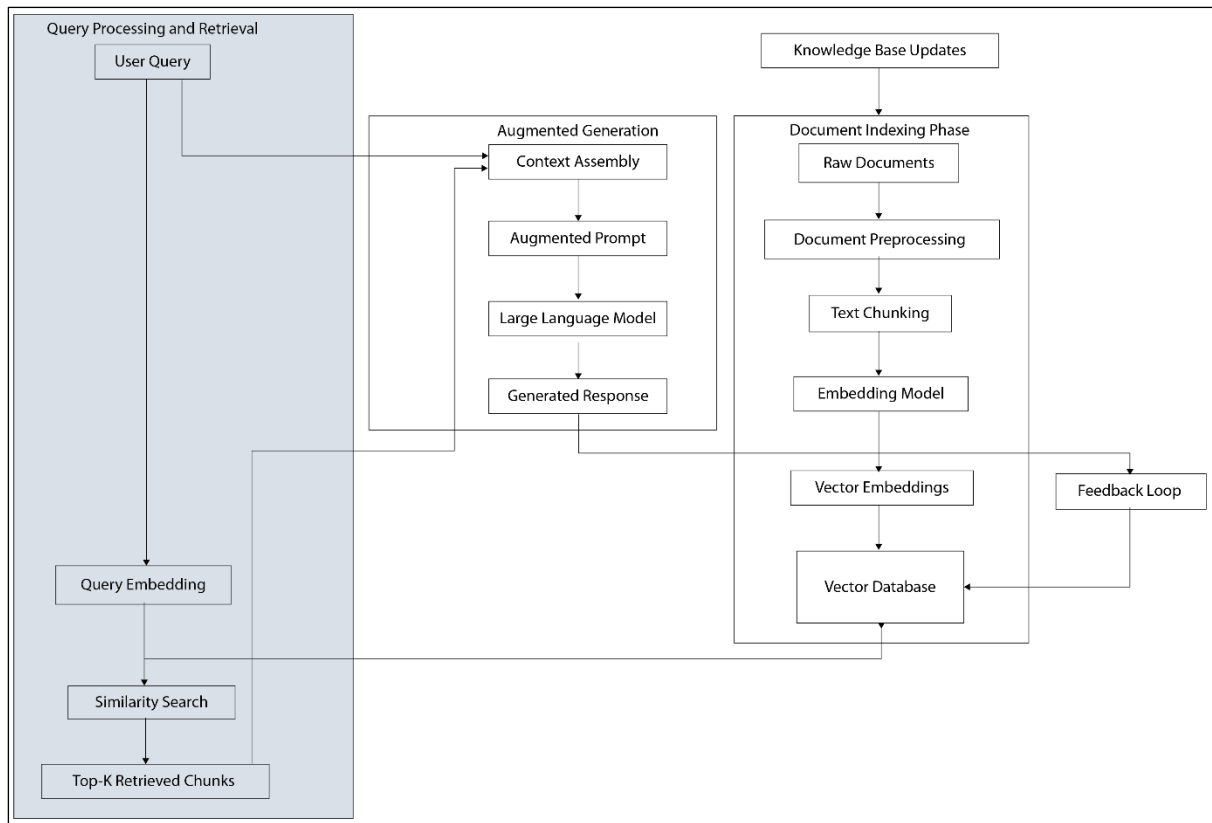
Preview

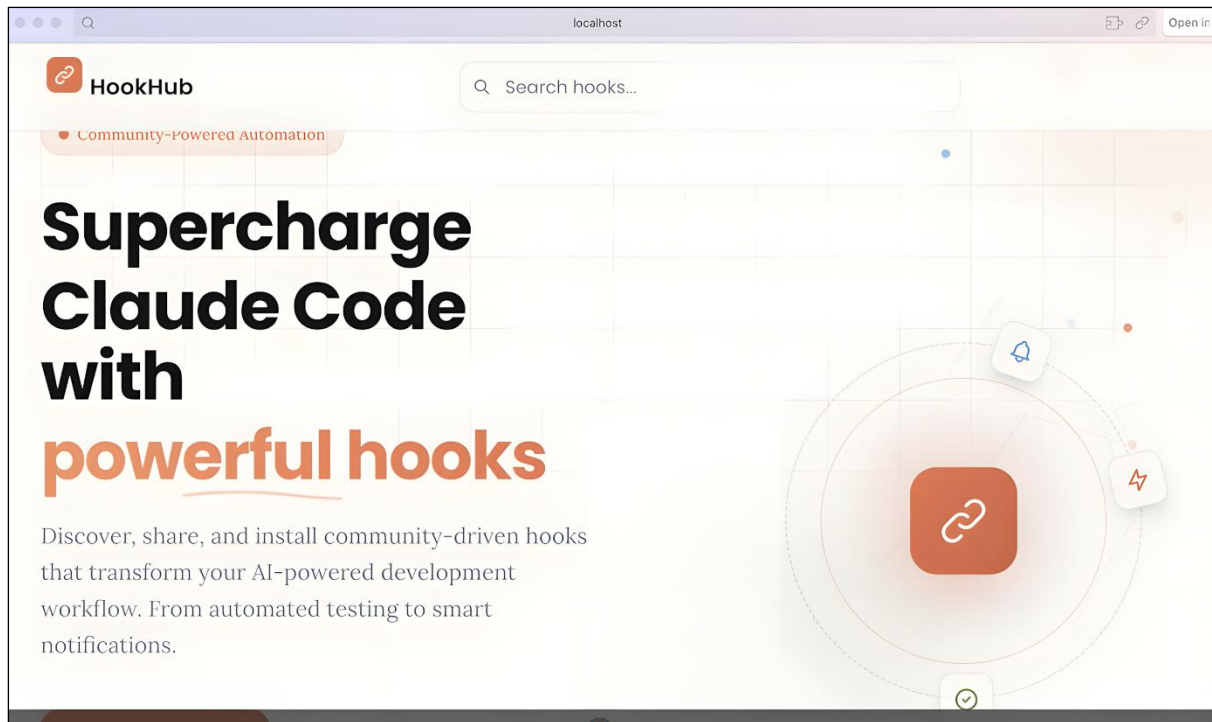
Error!

Parse error on line 25:
... classDef processBox fill:#e1f5fe,s
Expecting 'SPACE', 'AMP', 'COLON', 'DOWN', 'DEFAULT', 'NUM',
'COMMA', 'NODE_STRING', 'BRKT', 'MINUS', 'MULT',
'UNICODE_TEXT', got 'NEWLINE'

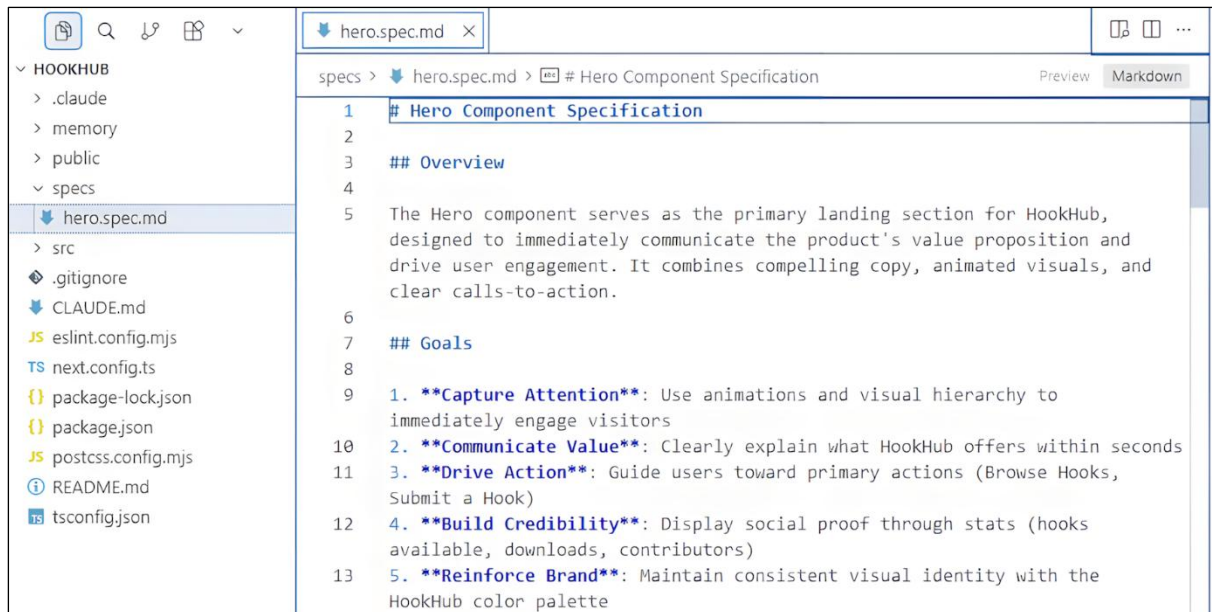
Insert →

Cmd Enter





```
Terminal  ...  node + -  [ ]  [X]  ...  ^  X
└─ Read 256 lines
● Bash(ls -la
  /Users/edenmarco/Desktop/claude-code-cra
  sh-course/hookhub/src/components/)
└─ total 40
  drwxr-xr-x  4 edenmarco  staff    128
  -rw-r--r--  1 edenmarco  staff    22 17:11 .
  ... +6 lines (ctrl+o to expand)
Add to Chat ⌘L
● I'll extract the Hero section into a new com
  ponent. Let me create Hero.tsx in the compon
  ts directory:
* Deliberating... (thought for 3s)
> █
Style: default Last Prompt: Hi, can you
```



Strategy for 6 New Variations:

1. **HeroMinimal** – Clean, stripped-down version with minimal animations
2. **HeroCentered** – Center-aligned layout without side visual
3. **HeroWaveform** – Audio/wave-inspired animated background
4. **HeroParticles** – Particle system background with constellation effect
5. **HeroTerminal** – Developer-focused with terminal/code aesthetic
6. **HeroGradientMesh** – Modern gradient mesh background with morphing shapes



Chapter 8: Creating and Customizing Output Styles

```
Claude Code ×
* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/edenmarco/GithubProjects/claude-code-crash-course

Choose your preferred output style:

This changes how Claude Code communicates with you

Use /output-style:new to create custom output styles

> 1. Default      Claude completes coding tasks efficiently and provides concise responses✓
   2. Explanatory  Claude explains its implementation choices and codebase patterns
   3. Learning     Claude pauses and asks you to write small pieces of code for hands-on practice

Enter to confirm • Esc to exit
```

```
Choose your preferred output style:

This changes how Claude Code communicates with you

Use /output-style:new to create custom output styles

> 1. Default      Claude completes coding tasks efficiently and provides concise responses
   2. Explanatory  Claude explains its implementation choices and codebase patterns
   3. Learning     Claude pauses and asks you to write small pieces of code for hands-on practice
   4. minimal-bullets Concise bullet-point responses with direct, no-fluff communication✓
```

```
Claude Code ×
> answer in bullet points, keep it concise and to the point. no fluff

● output-style-setup(Create concise bullet point style)
  | Done (5 tool uses • 18.3k tokens • 41.7s)

● • Created "minimal-bullets" output style
  • Located at: /Users/edenmarco/.claude/output-styles/minimal-bullets.md
  • Features: bullet points only, direct communication, no fluff
  • Ready for use in Claude Code

> /output-style
  | Set output style to minimal-bullets

> /output-style
  | Kept output style as minimal-bullets

> what are output styles?

● • Custom response formats for Claude Code
  • Control how Claude communicates and structures answers
  • Located in ~/.claude/output-styles/
  • Modify system prompts to change behavior
  • Examples: concise vs detailed, formal vs casual, bullet points vs paragraphs
  • Switch between styles using /output-style command
  • Create new ones with /output-style:new
```

Task: "Fix authentication bug"

Priority: "Urgent high"

Complexity: "Medium"

approach:

investigate:

auth_flow: &flow

- login_endpoint
- token_validation
- session_middleware

debugging:

logs: "./auth.log"

reproduce: "invalid_token_scenario"

fix:

dependencies: [*flow]

files_to_modify:

auth.js:

scope: "token refresh logic"

risk: "low"

middleware.js:

scope: "validation checks"

risk: "urgent"

verify:

tests: ["auth_integration", "token_expiry"]

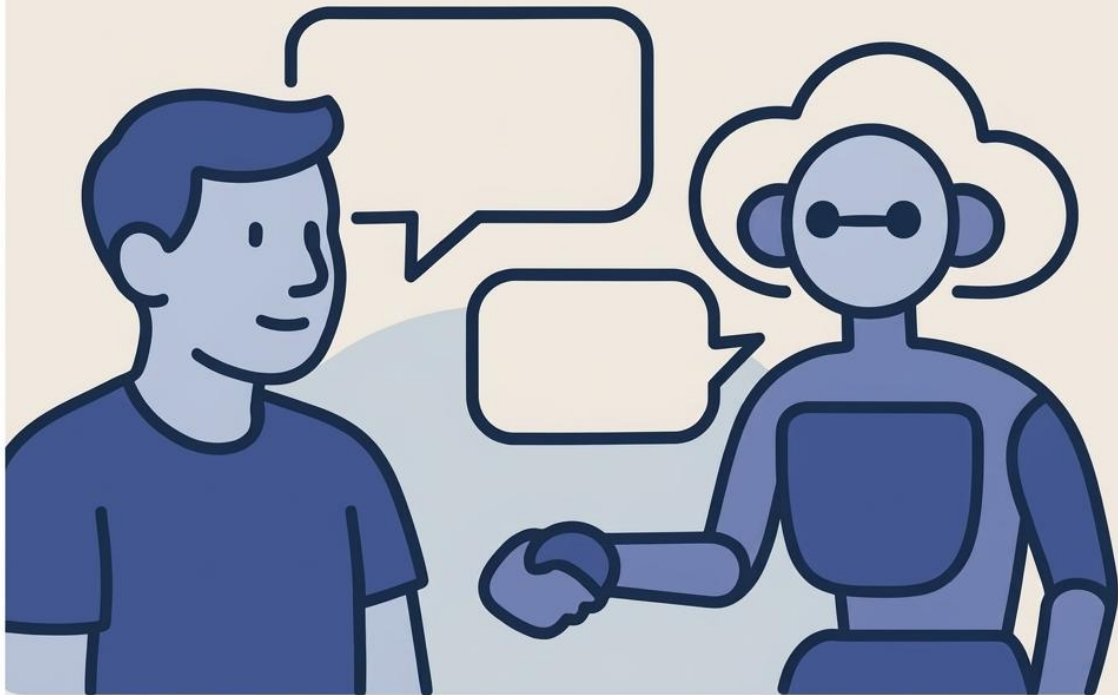
manual_check: "login flow end-to-end"

metadata:

estimated_time: "2 hours"

rollback_plan: "revert commits"

monitoring: "error_rate_dashboard"



HUMAN-AI COLLABORATION AND COMMUNICATION

☀ Claude Code ×

- Minimal explanations

> /output-style

└ Set output style to **yaml-concise**

> what are claude code output styles? give me an elaborate answer

- definition:

what: Custom response formats for Claude Code CLI

purpose: Control Claude's communication style and structure

scope: Global or project-level configuration

location:

Open folder in new window (cmd + click)

global: "~/.claude/output-styles/"

project: "{project}/.claude/output-styles/"

precedence: Project overrides global

file_structure:

format: Markdown with YAML frontmatter

frontmatter_required:

– description: Brief style explanation

content: System prompt additions/modifications

functionality:

controls:

– Response formatting (bullets, YAML, prose)

– Communication tone (formal, casual, technical)

– Information density (concise, detailed, comprehensive)

– Task approach (speed vs thoroughness)

– Code style preferences


```
edenmarco@Edens-MacBook-Pro-2 claud... % claude
* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/edenmarco/GithubProjects/claude-code-crash-course

> try "write a test for <filepath>"

? for shortcuts                                o /ide for Cursor
```

```
cwd: /Users/edenmarco/GithubProjects/claude-code-crash-course

Choose your preferred output style:

This changes how Claude Code communicates with you

Use /output-style:new to create custom output styles

1. Default          Claude completes coding tasks efficiently and provides
concise responses
> 2. Explanatory     Claude explains its implementation choices and
codebase patterns
3. Learning         Claude pauses and asks you to write small pieces of
code for hands-on practice
4. minimal-bullets  Concise bullet-point responses with direct, no-fluff
communication
5. retro-ascii-blog Format responses as retro HTML pages with ASCII art ✓
styling like a vintage blog
6. yaml-concise     Concise YAML-structured responses with minimal
explanations

Enter to confirm . Esc to exit
```

```
Last login: Sat Aug 23 17:42:40 on ttys012
edenmarco@Edens-MacBook-Pro-2 ~ % cd GithubProjects/claude-code-crash-course
edenmarco@Edens-MacBook-Pro-2 claud... % claude

* Welcome to Claude Code!

/help for help, /status for your current setup

cwd: /Users/edenmarco/GithubProjects/claude-code-crash-course

> /output-style
  L Set output style to minimal-bullets

>

? for shortcuts
```

```
Last login: Sat Aug 23 17:42:40 on ttys012
edenmarco@Edens-MacBook-Pro-2 ~ % cd GithubProjects/claude-code-crash-course
edenmarco@Edens-MacBook-Pro-2 claud... % claude

* Welcome to Claude Code!

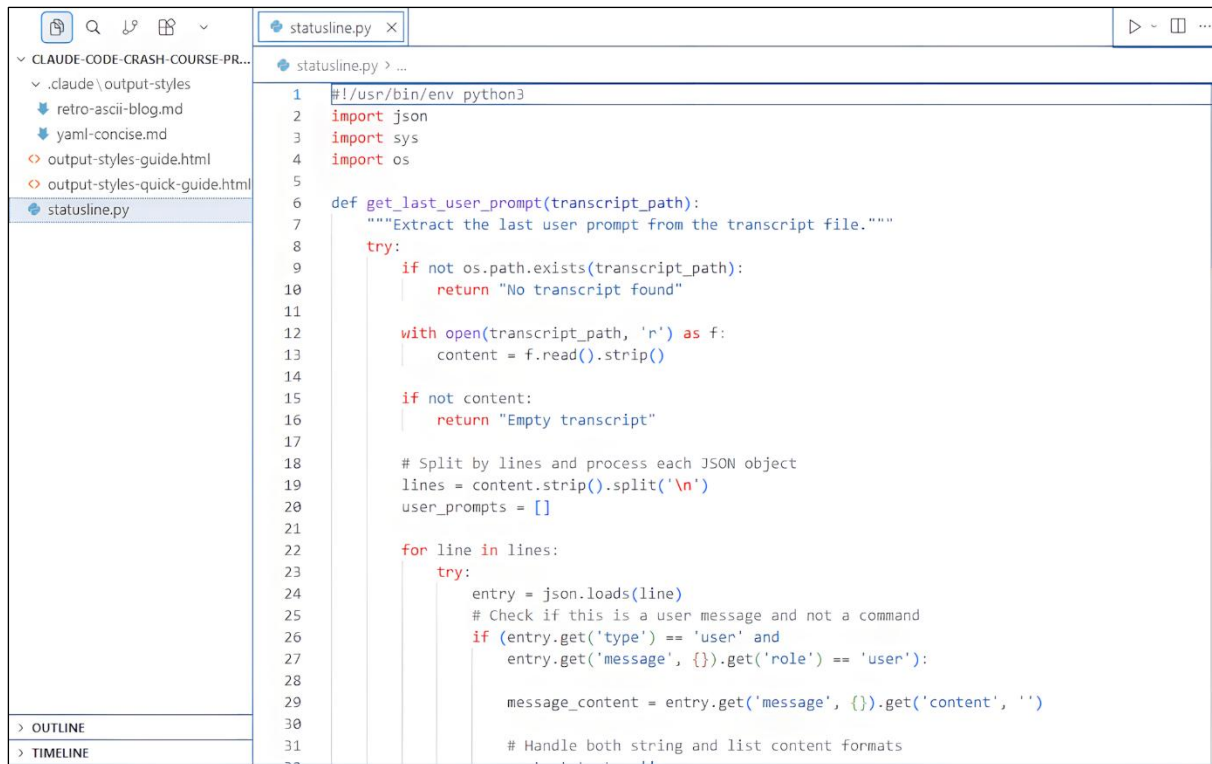
/help for help, /status for your current setup

cwd: /Users/edenmarco/GithubProjects/claude-code-crash-course

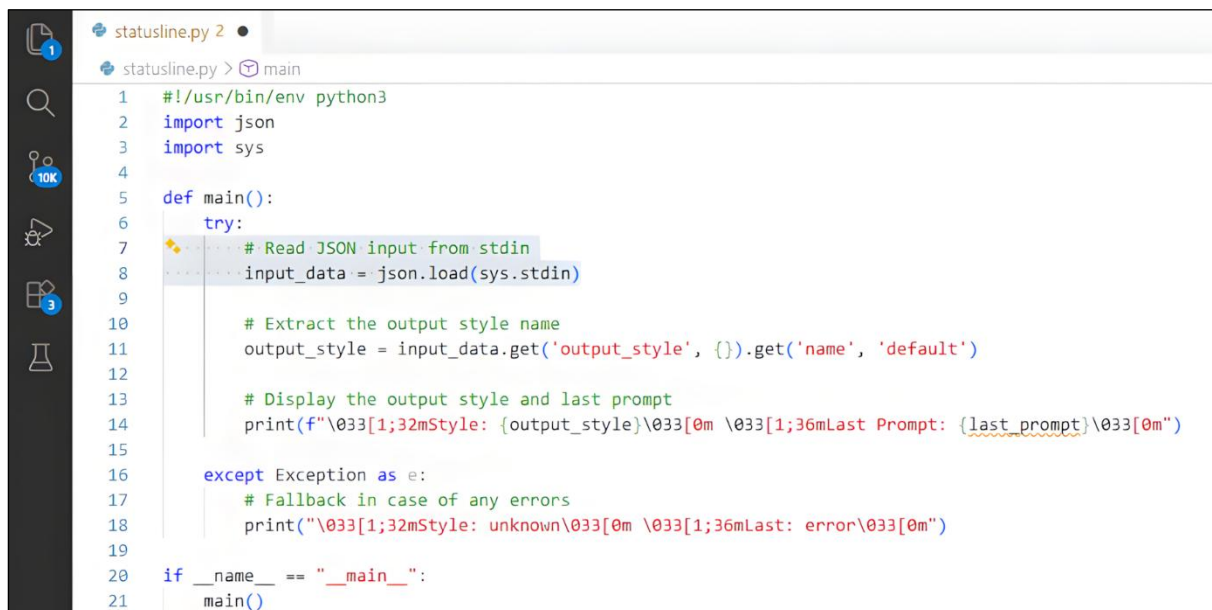
> /output-style
  L Set output style to yaml-concise

>

? for shortcuts
```



```
1 #!/usr/bin/env python3
2 import json
3 import sys
4 import os
5
6 def get_last_user_prompt(transcript_path):
7     """Extract the last user prompt from the transcript file."""
8     try:
9         if not os.path.exists(transcript_path):
10             return "No transcript found"
11
12         with open(transcript_path, 'r') as f:
13             content = f.read().strip()
14
15         if not content:
16             return "Empty transcript"
17
18         # Split by lines and process each JSON object
19         lines = content.strip().split('\n')
20         user_prompts = []
21
22         for line in lines:
23             try:
24                 entry = json.loads(line)
25                 # Check if this is a user message and not a command
26                 if (entry.get('type') == 'user' and
27                     entry.get('message', {}).get('role') == 'user'):
28
29                     message_content = entry.get('message', {}).get('content', '')
30
31                 # Handle both string and list content formats
```



```
1 #!/usr/bin/env python3
2 import json
3 import sys
4
5 def main():
6     try:
7         # Read JSON input from stdin
8         input_data = json.load(sys.stdin)
9
10        # Extract the output style name
11        output_style = input_data.get('output_style', {}).get('name', 'default')
12
13        # Display the output style and last prompt
14        print(f"\033[1;32mStyle: {output_style}\033[0m \033[1;36mLast Prompt: {last_prompt}\033[0m")
15
16    except Exception as e:
17        # Fallback in case of any errors
18        print("\033[1;32mStyle: unknown\033[0m \033[1;36mLast: error\033[0m")
19
20 if __name__ == "__main__":
21     main()
```

* Welcome to **Claude Code**!

/help for help, /status for your current setup

cwd: /Users/edenmarco/GithubProjects/claude-code-crash-course

> /output-style

└ Set output style to **minimal-bullets**

> /statusline is setting up statusline... i want you to create a statusline that displays the current used output-style. you should implement it in python and run it through uv

└ Allowed 3 tools for this command

● **statusline-setup**(Create statusline for output-style)

└ Done (5 tool uses · 4.6k tokens · 52.4s)

- Successfully configured statusline to display current output-style
- Created Python script at ~/.claude/statusline.py with uv execution
- Updated settings.json with statusLine configuration
- Status line now shows "Style: {output_style_name}" format

> can you make the style green and in bold text?

● **statusline-setup**(Make statusline green and bold)

└ Done (4 tool uses · 4.5k tokens · 20.9s)

- Updated statusline script to display green and bold text
- Added ANSI color codes: \033[1;32m for bold green
- Status line now shows "Style: {output_style}" in **bold green** format

> █



Style: **minimal-bullets**

▶▶ accept edits on (shift+tab to cycle)

Name	Date Modified	Size	Kind
> -Users-edenmarco-Githu...laude-code-crash-course	Today, 12:43	--	Folder
> -Users-edenmarco	Yesterday, 21:20	--	Folder
> -Users-edenmarco-Desktop	Yesterday, 21:18	--	Folder
> -Users-edenmarco-Desktop-langchain-course	23 August 2025, 20:44	--	Folder
> -Users-edenmarco-Desktop-ice-breaker	16 August 2025, 10:39	--	Folder
> -Users-edenmarco-Desktop-IceBreaker	15 August 2025, 20:20	--	Folder
> -Users-edenmarco-Desktop-sampling	14 August 2025, 11:59	--	Folder
> -Users-edenmarco-Githu...de-crash-course-hookhub	13 August 2025, 12:41	--	Folder
> -Users-edenmarco-GithubProjects-cursor-book	6 August 2025, 11:29	--	Folder
> -Users-edenmarco-Desktop-planning	2 August 2025, 13:12	--	Folder
> -Users-edenmarco-Desktop-pricing	2 August 2025, 8:41	--	Folder
> -Users-edenmarco-Desktop-tmp3-ice-breaker	29 July 2025, 15:01	--	Folder
> -Users-edenmarco-Desktop-tmp2-ice-breaker	29 July 2025, 14:06	--	Folder

> {"role": "user"

the messages below were generated by

```
{ "role": "user", "content": "<command-na  
: { "role": "user", "content": "<local-comm  
' : { "role": "user", "content": "hello" }, "uui  
lWcL17Gsp8EfYWCumDEYoB2", "type": "message
```

```
statusline.py X
statusline.py > main
6 def get_last_user_prompt(transcript_path):
50
51     if user_prompts:
52         # Return the last user prompt, truncated if too long
53         last_prompt = user_prompts[-1]
54         if len(last_prompt) > 50:
55             return last_prompt[:47] + "..."
56         return last_prompt
57     else:
58         return "No user prompts found"
59
60 except Exception as e:
61     return f"Error reading transcript: {str(e)}"
62
63 def main():
64     try:
65         # Read JSON input from stdin
66         input_data = json.load(sys.stdin)
67
68         # Extract the output style name
69         output_style = input_data.get('output_style', {}).get('name', 'default')
70
71         # Extract the transcript path
72         transcript_path = input_data.get('transcript_path', '')
73
74         # Get the last user prompt
75         last_prompt = get_last_user_prompt(transcript_path)
76
77         # Display the output style and last prompt
78         print(f"\033[1;32mStyle: {output_style}\033[0m \033[1;36mLast Prompt: {last_prompt}\033[0m")
79
```

```
startswith() on a list object instead of a string. This happens when the
message content is multimodal (contains text blocks, images, etc.).

The statusline agent has updated the code to properly handle both:
- Simple string messages
- Complex multimodal messages with content blocks

The status line should now work correctly without the "'list' object has
no attribute 'startswith'" error.

> hello!

• Hello! 🌟 What can I help you with today?

> hows it going?

• Going well! Ready to help with any coding tasks you have. What would you
like to work on?

> /output-style
  ↳ Set output style to yaml-concise

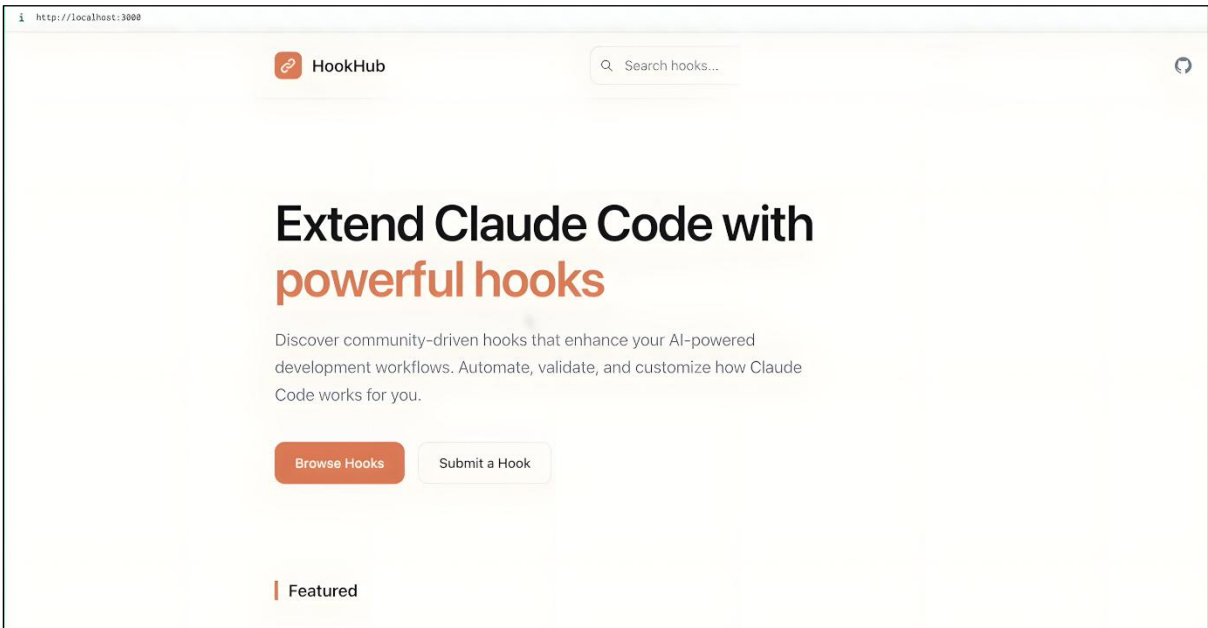
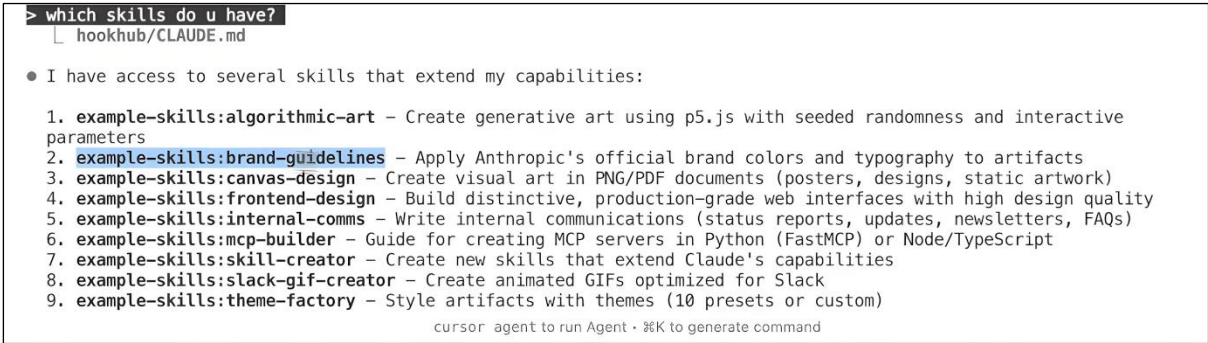
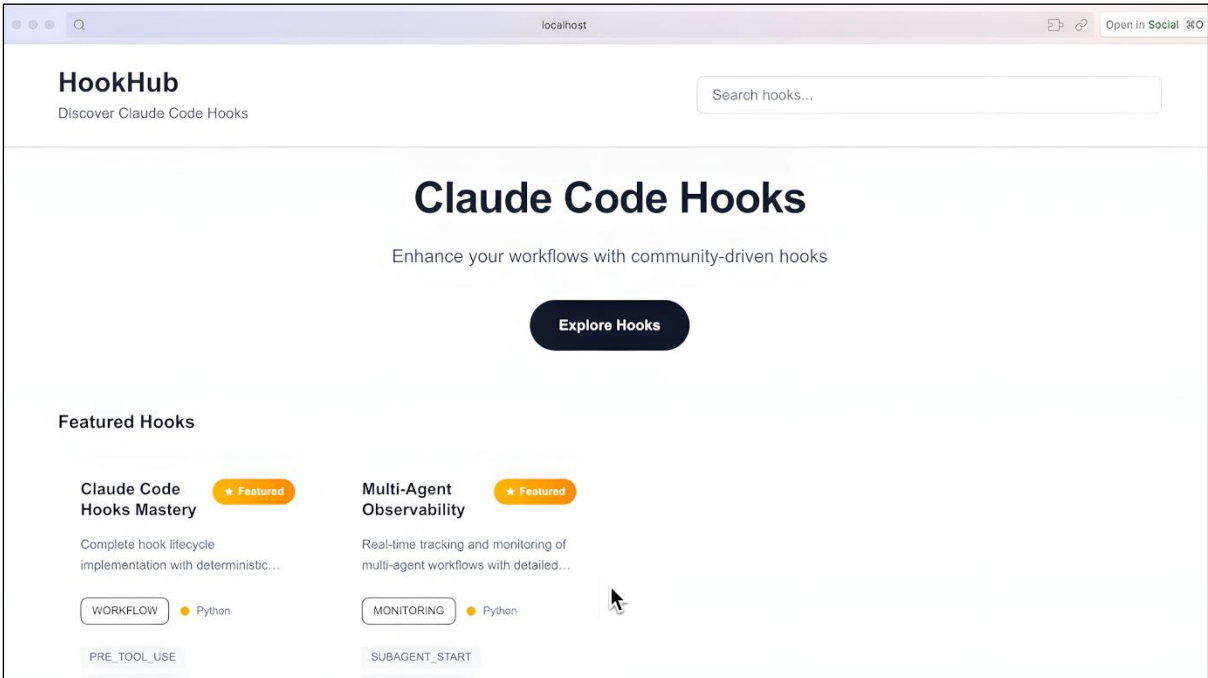
> |

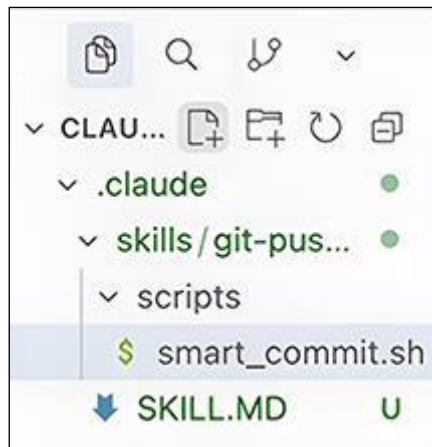
Style: yaml-concise Last: hows it going?
▶▶ accept edits on (shift+tab to cycle)
```

```
Users > edenmarco > claude > projects > -Users-edenmarco-GithubProjects-claude-code-crash-
1 {"parentUid": "> message_content", "isSidechain": false, "u
2 {"parentUid": "6fe515-545d-4885-9504-80272f4715e0", "isSidechain": false, "u
3 {"parentUid": "81b72f-0bf5-4a23-837e-4c8ce48e2fe1", "isSidechain": false, "u
4 {"parentUid": "42dd8b-f016-4935-ba21-24e5c1591730", "isSidechain": false, "u
5 {"parentUid": "19a2b8-20fc-453a-a72f-35d81bd1c17c", "isSidechain": false, "u
6 {"parentUid": "fad66b-2bb3-4b47-bffe-198d06cbb39c", "isSidechain": false, "u
7 {"parentUid": "bdb19e-0c16-4821-a13f-2aeece3f1f0", "isSidechain": false, "u
8 {"parentUid": null, "isSchain": true, "userType": "external", "caw": "/Users/e
9 {"parentUid": "ebdb68-8386-4123-af6b-836c7be7db47", "isSidechain": true, "us
10 {"parentUid": "4761c5-c75b-4f6e-90b3-18d708540254", "isSidechain": true, "us
11 {"parentUid": "33aaf5de-14-4c09-8564-bfa6151c3f5c", "isSidechain": true, "us
12 {"parentUid": "a22cdd8f-2162-495a6c-2693524548ed", "isSidechain": true, "us
13 {"parentUid": "20af0ce8-777c-42595-bas1d5e4a2288", "isSidechain": true, "us
14 {"parentUid": "7b5a0f09-0494-40c3fb-81571f115491", "isSidechain": true, "us
15 {"parentUid": "4edde19c-31b5-4cdf-fa365c0946d1", "isSidechain": true, "us
16 {"parentUid": "ca98cafd-825a-46a7-se66fdf1ef952", "isSidechain": true, "us
17 {"parentUid": "75eef4c3-2dff-47a5-566361bb8340c", "isSidechain": true, "us
18 {"parentUid": "142b12a9-cbdc-43d0-6accd313baae", "isSidechain": true, "us
19 {"parentUid": "c804ce60-b56c-4864-4c1cd991e8fc9", "isSidechain": false, "u
20 {"parentUid": "e2ee1f8a-1c02-4b70-a85733e8fc849", "isSidechain": false, "u
21 {"parentUid": "5fa52b5d-3a0f-497a-d8e683d33563", "isSidechain": false, "u
22
23
```



Chapter 9: Understanding Agent Skills





When to Use

Automatically activate when the user:

- Explicitly asks to push changes ("push this", "commit and push", "push to remote")
- Mentions saving work to remote ("save to github", "push to remote")
- Completes a feature and wants to share it
- Says phrases like "let's push this up" or "commit these changes"

```
---
name: git-pushing
description: Stage, commit, and push git changes with conventional commit messages. Use when user wants to commit and push changes, mentions pushing to remote, or asks to save and push their work. Also activates when user says "push changes", "commit and push", "push this", "push to github", or similar git workflow requests.
---
```

Git Push Workflow

Stage all changes, create a conventional commit, and push to the remote branch.

When to Use

Automatically activate when the user:

- Explicitly asks to push changes ("push this", "commit and push", "push to remote")
- Mentions saving work to remote ("save to github", "push to remote")
- Completes a feature and wants to share it
- Says phrases like "let's push this up" or "commit these changes"

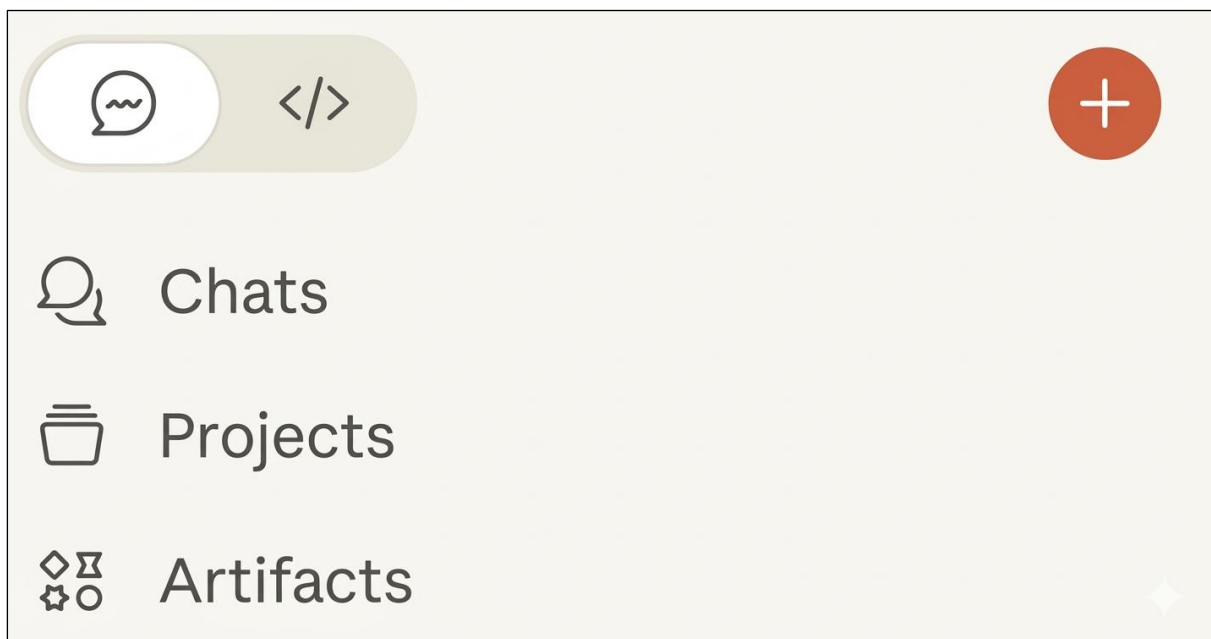
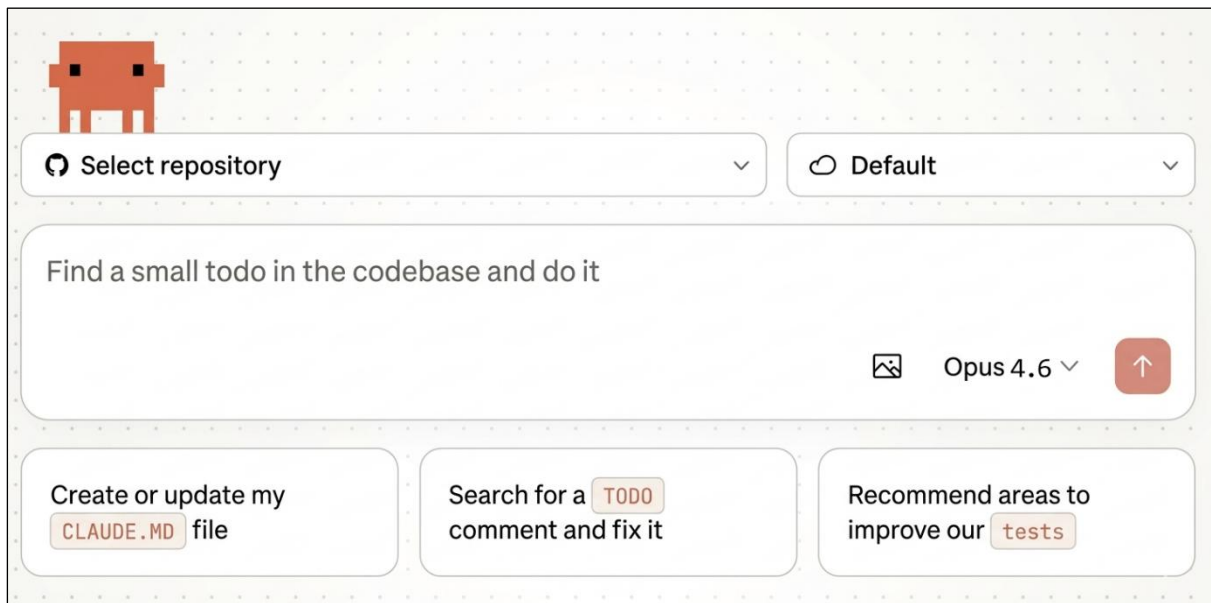
```
if [ -n "$SCOPE" ]; then
    COMMIT_MSG="${COMMIT_TYPE}(${SCOPE}): ${DESCRIPTION}"
else
    COMMIT_MSG="${COMMIT_TYPE}: ${DESCRIPTION}"
fi

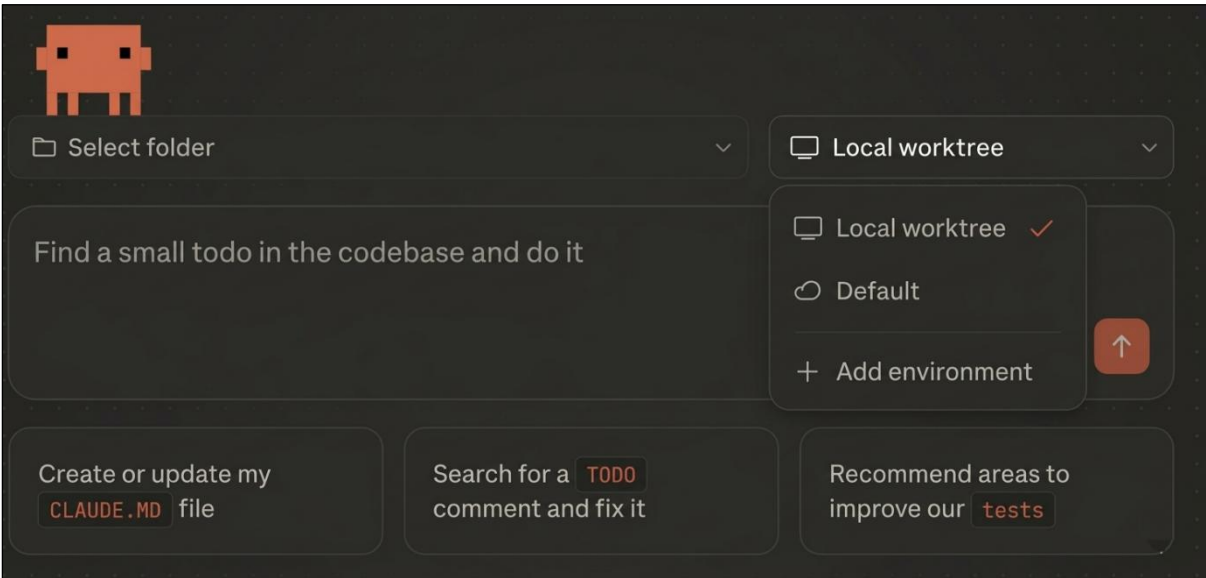
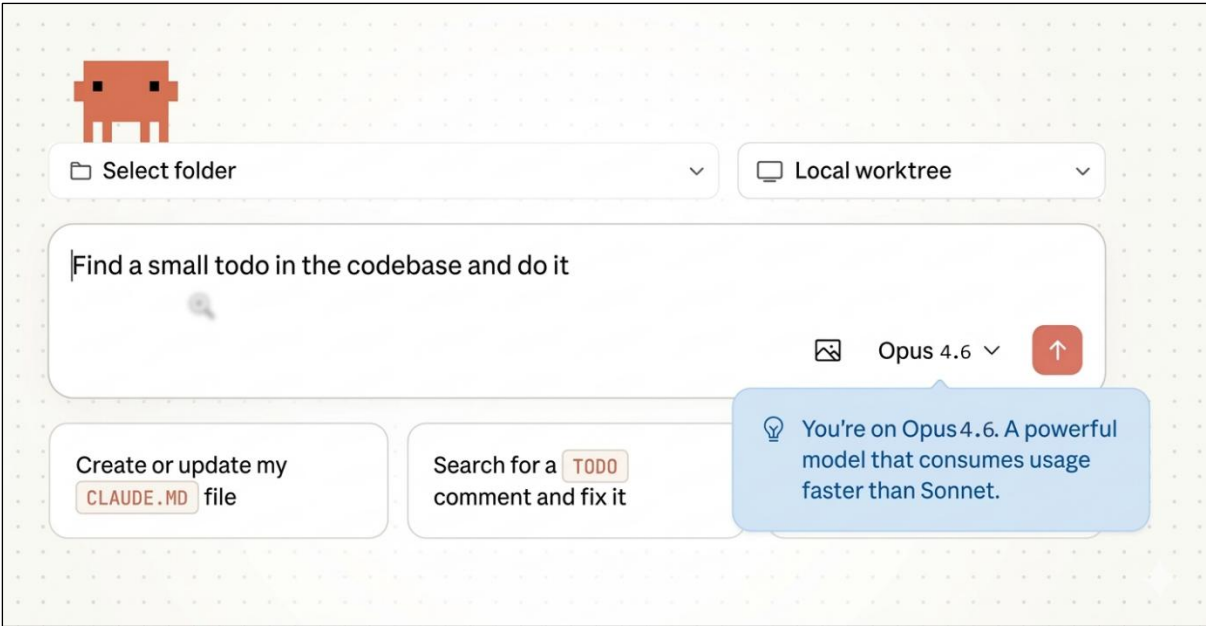
info "Generated commit message: $COMMIT_MSG"
else
    COMMIT_MSG="$1"
    info "Using provided message: $COMMIT_MSG"
fi

# Create commit with Claude Code footer
git commit -m "$(cat <<EOF
${COMMIT_MSG})
```

Add to Chat  Quick Edit 

Chapter 10: Using Claude Code Desktop







Select repository

Default

Search repositories

tovana
assafelovic

ai-code-interpreter
emarco177

ai-engineering-jobs
emarco177

AnswerNator-FrontEnd
emarco177

Awesome-Black-Friday-Cyber-Monday
emarco177

claude-code-crash-course
emarco177

..



Opus 4.5



Recommend areas to
improve our **tests**

Git Worktrees

One repo, multiple working directories

📁 ~/projects/

📁 ONE .git repository

👤 MAIN WORKTREE

main branch

- 📁 my-app/
 - 📁 .git/
 - 📁 src/
 - 📄 App.tsx
 - 📁 components/
 - 📄 package.json

🛠️ AGENT WORKTREE

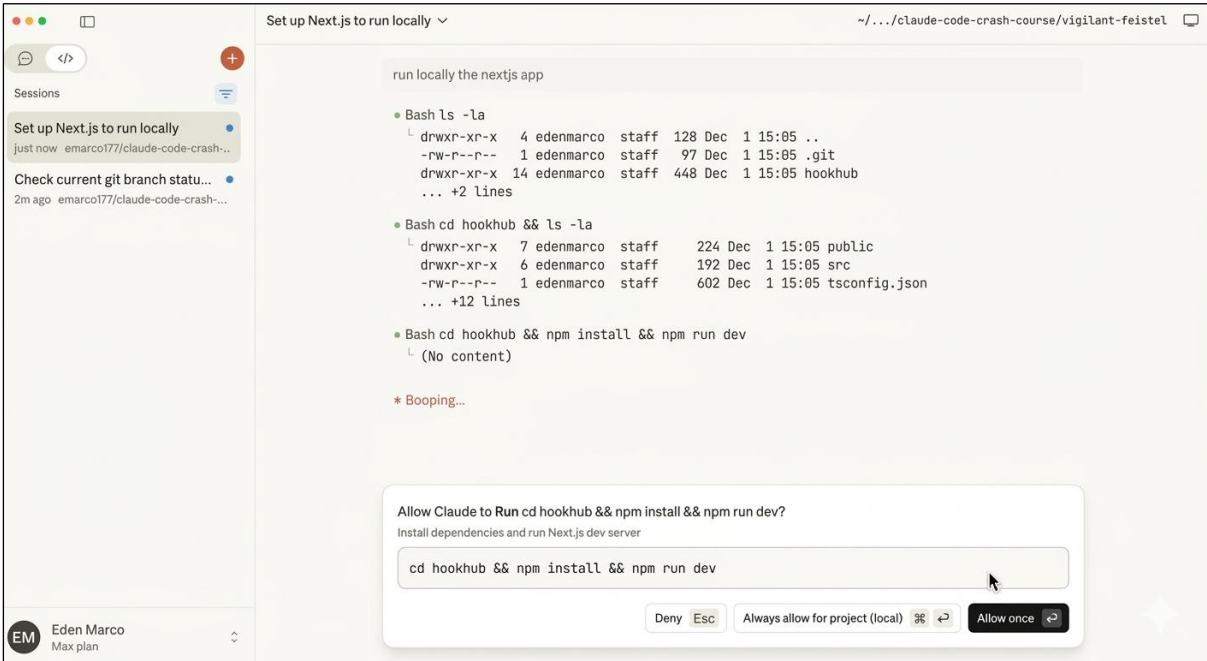
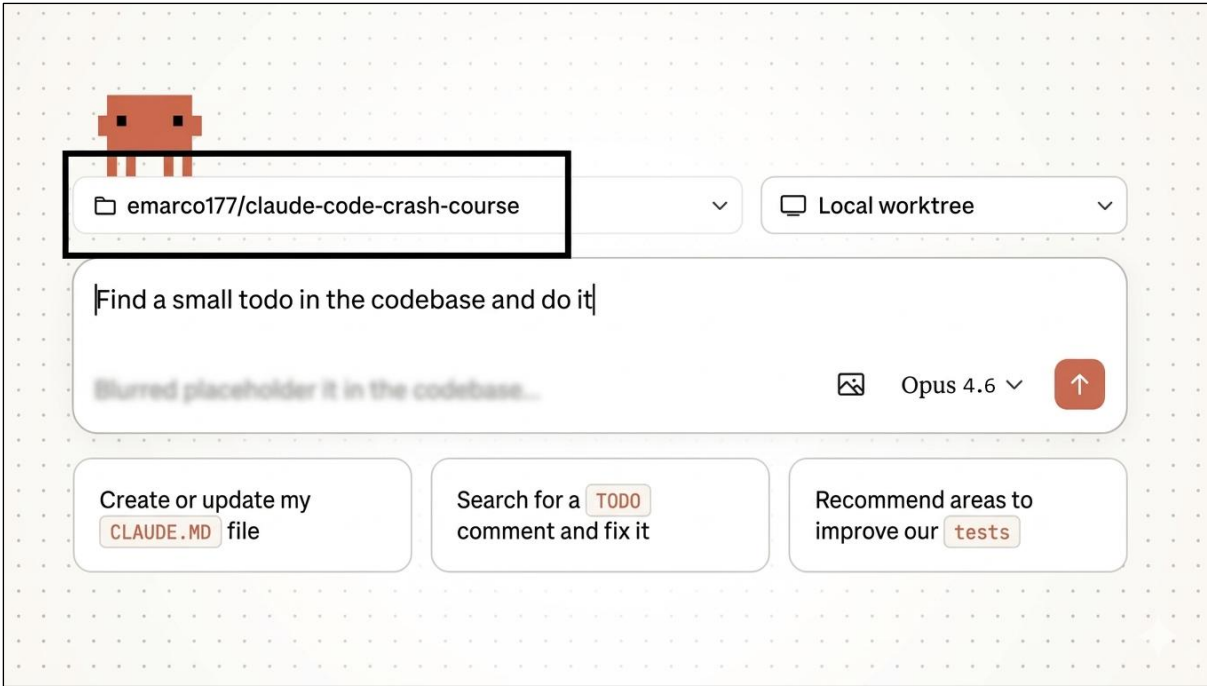
feature branch

- 📁 my-app-feature/
 - linked to my-app/.git
 - 📁 src/
 - 📄 App.tsx
 - 📄 NewFeature.tsx
 - 📁 components/
 - 📄 package.json

🛠️ ANOTHER WORKTREE

hotfix branch

- 📁 my-app-hotfix/
 - linked to my-app/.git
 - 📁 src/
 - 📄 ...



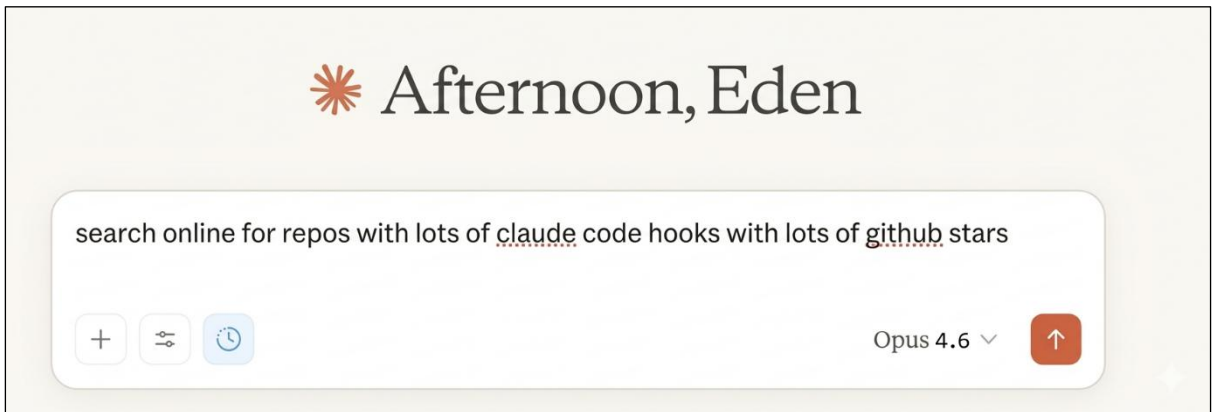
Set up Next.js to run locally ▾ ~/.../claude-code-crash-course/vigilant-feistel

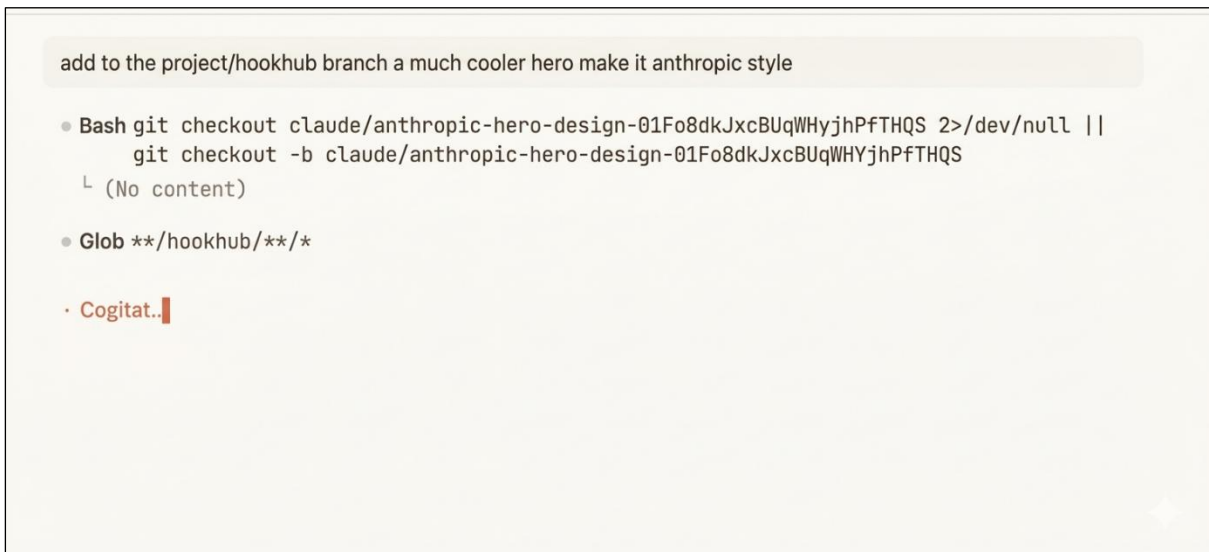
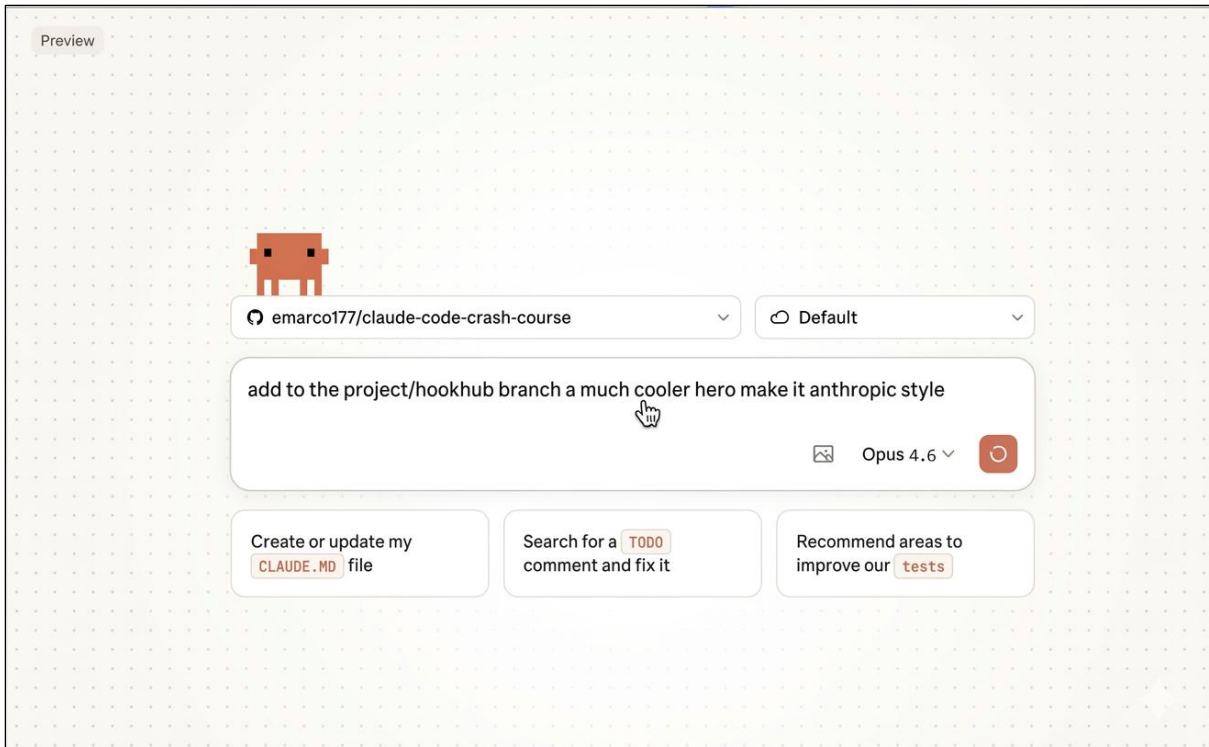
↳ Command running in background with ID: adc9f5

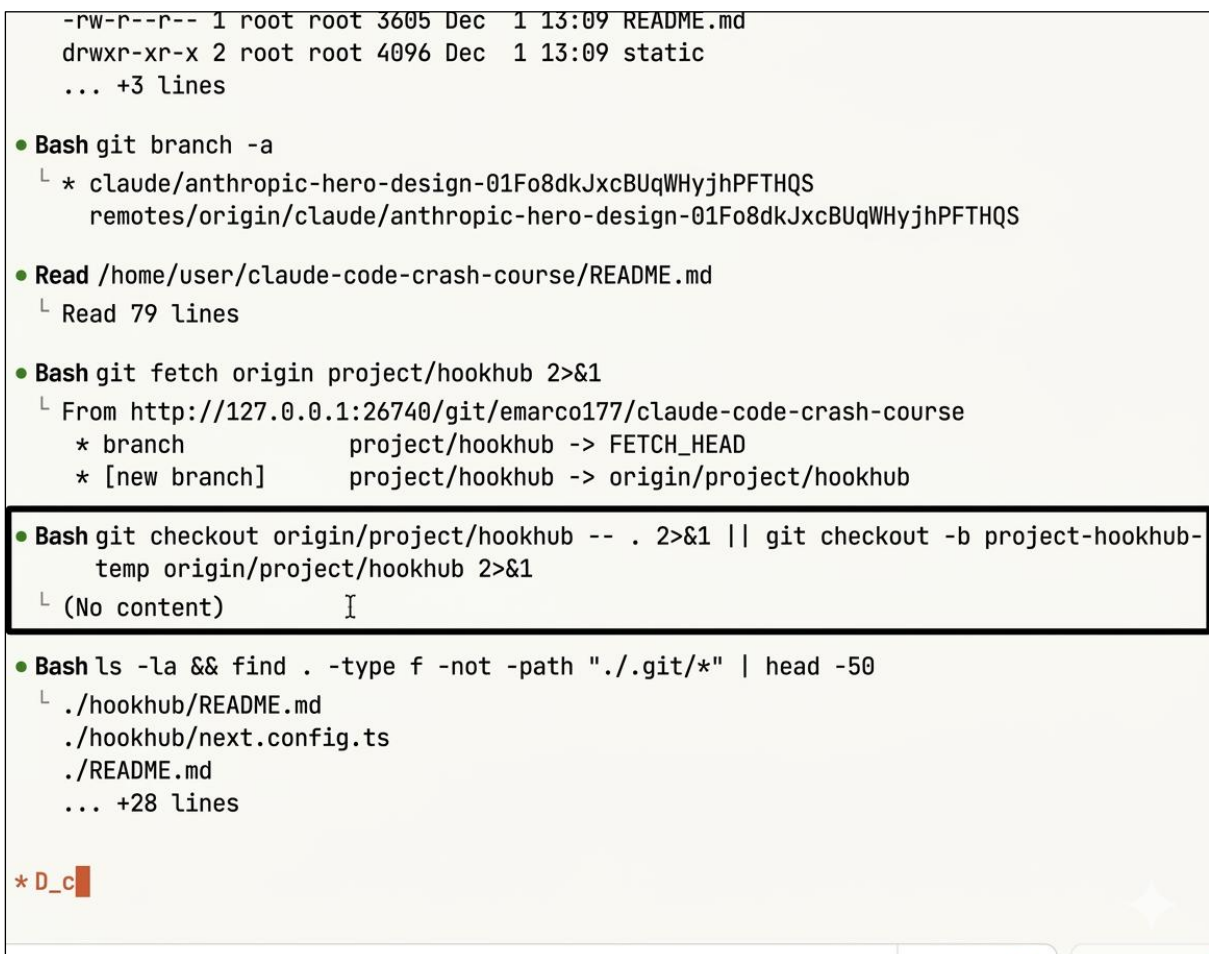
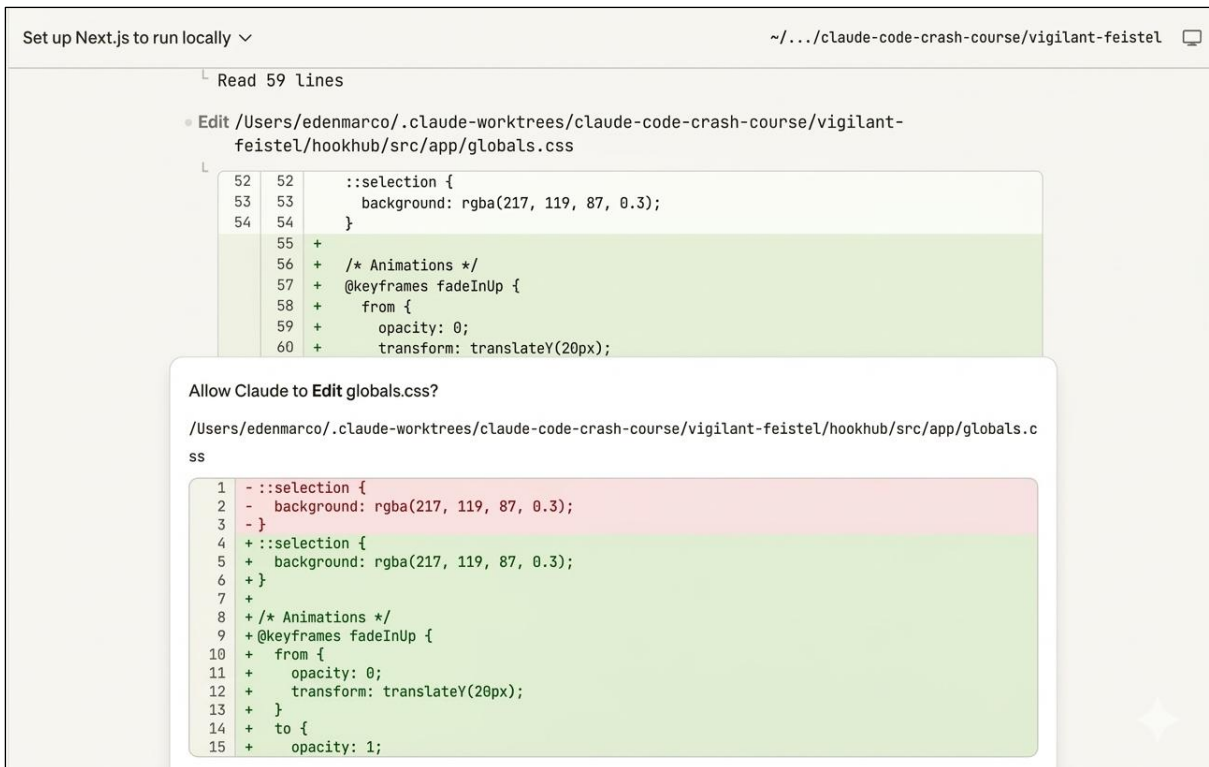
- **BashOutput**
- **Bash** which node && which npm
 - ↳ Exit code 1
 - node not found
- **Bash** source ~/.nvm/nvm.sh && which npm && npm --version
 - ↳ /Users/edenmarco/.nvm/versions/node/v22.18.0/bin/npm
 - 11.5.2
- **Bash** source ~/.nvm/nvm.sh && npm install && npm run dev
 - ↳ Command running in background with ID: 47efb7
- **BashOutput**
- **BashOutput**
- **BashOutput**
- **BashOutput**
- The Next.js app is now running locally at <http://localhost:3001> (port 3000 was in use, so it's using 3001 instead).
The server is ready and you can access the HookHub app in your browser.

VS Code

CLI







disler claude-code-hooks github stars repository

10 results

Curated popular repositories featuring Claude Code hooks.

Here are the top GitHub repositories with Claude Code hooks that have significant stars:

Top Claude Code Hooks Repositories

1. **awesome-claude-code** by hesreallyhim

- **Stars:** ~17.4k ★
- **URL:** <https://github.com/hesreallyhim/awesome-claude-code>
- A curated list of awesome commands, files, and workflows for Claude Code [GitHub](#). This is the definitive collection that includes hooks, slash commands, CLI tools, CLAUDE.md files, and more.

2. **claude-code-hooks-mastery** by disler (IndyDevDan)

- **Stars:** ~1.9k ★
- **URL:** <https://github.com/disler/claude-code-hooks-mastery>
- Demonstrates how to use Claude Code hooks to add deterministic (or non-deterministic) control over Claude Code's behavior. [GitHub](#) Covers all 8 hook lifecycle events with complete implementations including security filtering, TTS feedback, and sub-agents.

with Skills, with a technique that leverages hooks to ensure Claude intelligently selects and activates the appropriate Skill given the current context. [GitHub](#)

Notable Mentions from awesome-claude-code:

claude-code-tools by Prasad Chalasani - tmux integrations, better session management, security-enhancing hooks

ClaudoPro Directory - Wide selection of hooks, slash commands, and subagent files

TypeScript Quality Hooks by bartolli - Quality check hooks with SHA256 config caching for <5ms validation

add those links to the hooks list

URL: <https://github.com/johnmduquist/claude-hooks> typescript-powered hook system for Claude Code with full type safety and auto-completion. GitHub Provides strongly-typed payloads for all hook types (PreToolUse, PostToolUse, Notification, Stop).

6. Claude Code Infrastructure Showcase by diet103

Mentioned in awesome-claude-code as a remarkably innovative approach to working with Skills, with a technique that leverages hooks to ensure Claude intelligently selects and activates the appropriate Skill given the current context. GitHub

Notable Mentions from awesome-claude-code:

claude-code-tools by Prasad Chalasani - tmux integrations, better session management, security-enhancing hooks
ClaudePro Directory - Wide selection of hooks, slash commands, and subagent files
TypeScript Quality Hooks by bartolli - Quality check hooks with SHA256 config caching for <5ms validation

add those links to the hooks list

- I'll read the current hooks data file to understand its structure, then add the new repositories.

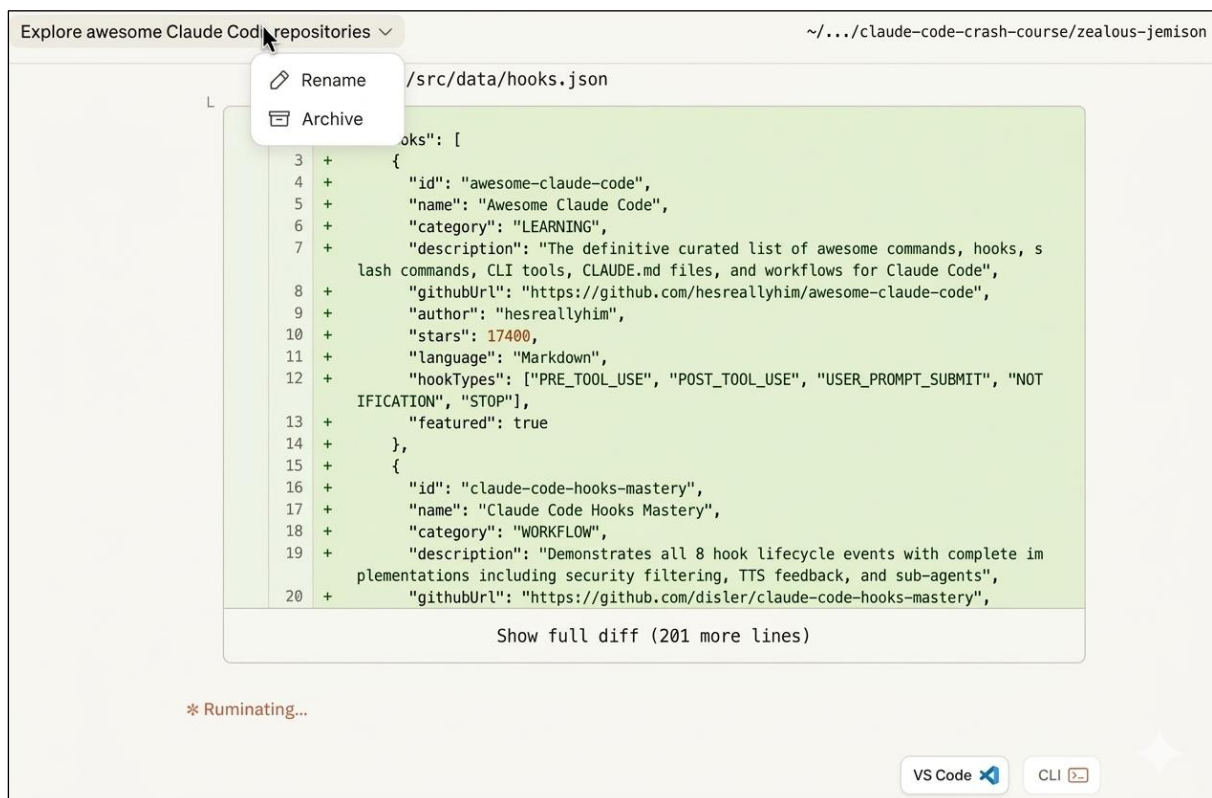
- **Glob** `**/*hooks*`

- **Glob** `**/data/**`

- **Read** `/Users/edenmarco/.claude-worktrees/claude-code-crash-course/zealous-jemison/hookhub/src/data/hooks.json`

└─ Read 152 lines

- Now I'll update the hooks.json file with the new repositories. I'll update the existing entries with correct star counts and add the new ones:



- Done! I've added 6 new repositories and updated 2 existing ones:

New entries added:

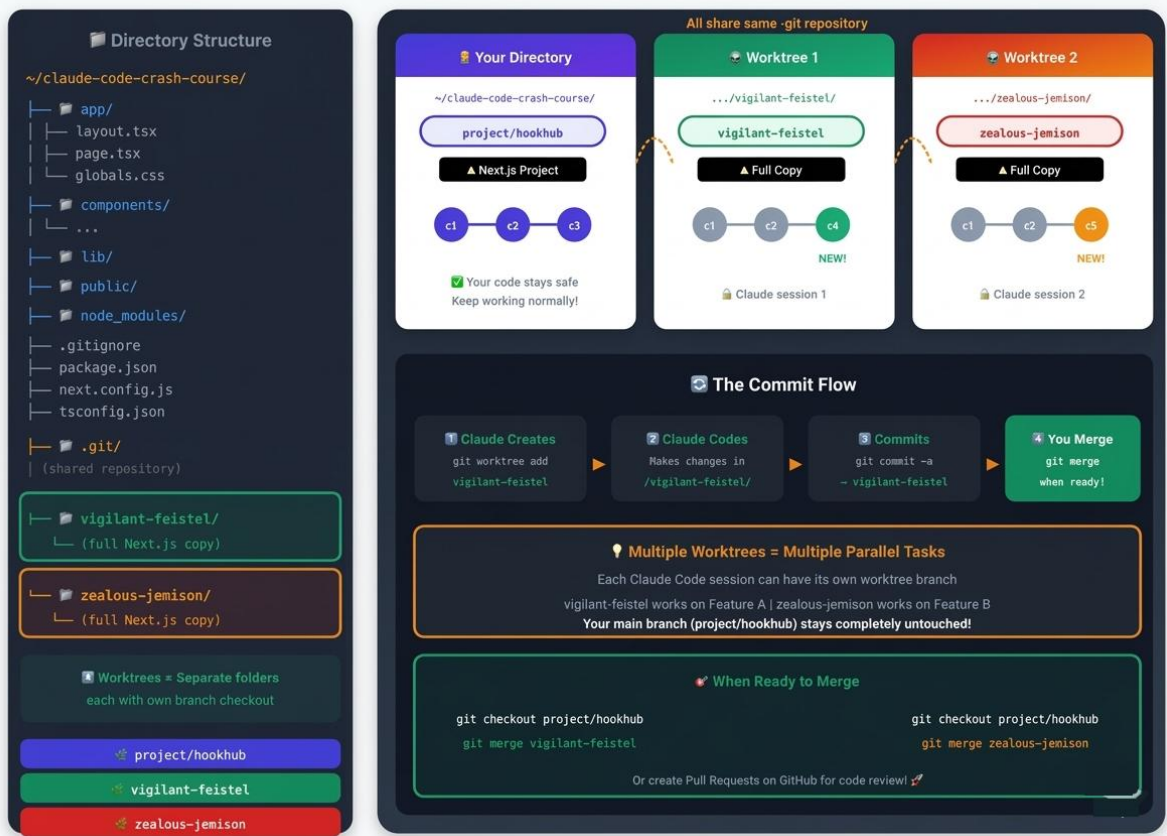
1. **Awesome Claude Code** (17.4k stars) - The definitive curated list by hesreallyhim
2. **ClaudeKit** (3.2k stars) - CLI toolkit with 20+ subagents by carlrannaberg
3. **Claude Hooks (TypeScript)** (500 stars) - Type-safe hooks by johnlindquist
4. **Claude Code Infrastructure Showcase** (350 stars) - Skills/hooks integration by diet103
5. **Claude Code Tools** (280 stars) - Tmux integrations by pchalsani
6. **TypeScript Quality Hooks** (180 stars) - Quality validation by bartolli

Updated existing entries:

- **Claude Code Hooks Mastery** - Updated stars (234 → 1,900) and description to mention all 8 hook types
- **Multi-Agent Observability** - Updated stars (189 → 803) and improved description

The list is now ordered by star count (highest first) with the top 5 most popular repos marked as featured. ✨

Git Worktrees: Claude Code + Next.js Project



feistel/hookhub/src/components/HookCard.tsx

```
L
29 29      return (
30      -      <div className="group bg-[var(--background)] border border-[var(--border)] rounded-xl p-5 hover:border-[#d97757]/40 hover:shadow-lg hover:shadow-[#d97757]/5 transition-all duration-200">
30      +      <div className="group bg-[var(--background)] border border-[var(--border)] rounded-xl p-5 hover:border-[#d97757]/40 hover:shadow-lg hover:shadow-[#d97757]/5 transition-all duration-300 hover:-translate-y-1">
```

• BashOutput

- The animations are now live. Refresh <http://localhost:3001> to see the changes:

Animations added:

1. Hero section:

- Title fades in and slides up
- Description and buttons have staggered fade-in-up animations
- "Browse Hooks" button has a subtle pulsing glow effect
- Both buttons scale up on hover

2. Logo: Floating animation in the header

3. Section headers: Slide in from the left with staggered delays

4. Hook cards:

- Staggered fade-in-up animations when the page loads
- Cards lift up slightly on hover (`hover:-translate-y-1`)
- Enhanced shadow on hover

Allow Claude to Run `git add hookhub/src/app/globals.css hookhub/src/app/page.tsx`

`hookhub/src/components/HookCard.tsx` && `git commit -m "$(cat <<'EOF' Add animations to landing page - Add fade-in-up, slide-in-left, and float animations - Hero section animates in with staggered delays - Browse Hooks button has pulsing glow effect - Logo floats in header - Hook cards animate in with staggered timing and lift on hover - Section headers slide in from left Via Claude Code Desktop 🧙 Generated with [Claude Code] (https://claude.com/claude-code) Co-Authored-By: Claude <noreply@anthropic.com> EOF)"`?

Stage and commit animation changes

```
git add hookhub/src/app/globals.css hookhub/src/app/page.tsx hookhub/src/components/Ho
Add animations to landing page
```

- Add fade-in-up, slide-in-left, and float animations
- Hero section animates in with staggered delays
- Browse Hooks button has pulsing glow effect
- Logo floats in header
- Hook cards animate in with staggered timing and lift on hover
- Section headers slide in from left

Via Claude Code Desktop

Deny Esc

Always allow for project (local) ⌘ ↵

Allow once ↵

- **Bash git diff**

```
└─ <div className="flex items-start justify-between gap-3 mb-3">
    <h3 className="text-base font-semibold text-[var(--foreground)] leading-
snug">...
... +180 lines
```

- **Bash git log --oneline -5**

```
└─ 16a2531 initial project
   fa598ec 1-start-here
   1f4813c first commit
   ... +2 lines
```

- **Bash git add hookhub/src/app/globals.css hookhub/src/app/page.tsx**

```
hookhub/src/components/HookCard.tsx && git commit -m "$(cat <<'EOF' Add animations
to landing page - Add fade-in-up, slide-in-left, and float animations - Hero
section animates in with staggered delays - Browse Hooks button has pulsing glow
effect - Logo floats in header - Hook cards animate in with staggered timing and
lift on hover - Section headers slide in from left Via Claude Code Desktop 🌐
Generated with [Claude Code](https://claude.com/claude-code) Co-Authored-By:
Claude <noreply@anthropic.com> EOF )"
└─ [vigilant-feistel b438f44] Add animations to landing page
   3 files changed, 100 insertions(+), 13 deletions(-)
```

* Musing...

- **Bash git add hookhub/src/data/hooks.json && git commit -m "\$(cat <<'EOF' Add top Claude Code hooks repositories to hooks list Added 6 new repositories from the community:**
 - Awesome Claude Code (17.4k stars) - curated list by hesreallyhim - ClaudeKit (3.2k stars) - CLI toolkit by carlrannaberg - Claude Hooks TypeScript (500 stars) - type-safe hooks by johnlindquist - Claude Code Infrastructure Showcase - Skills integration by diet103 - Claude Code Tools - tmux integrations by pchalsani - TypeScript Quality Hooks - quality validation by bartolli Updated star counts and descriptions for existing entries. Via Claude Code Desktop 🌐 Generated with [Claude Code](https://claude.com/claude-code) Co-Authored-By: Claude <noreply@anthropic.com> EOF)"

```
└─ [zealous-jemison d88a8b3] Add top Claude Code hooks repositories to hooks list
   1 file changed, 79 insertions(+), 7 deletions(-)
```

- **Bash git status**

```
└─ On branch zealous-jemison
   nothing to commit, working tree clean
```

- Committed successfully. The commit d88a8b3 includes the 6 new hooks repositories and updated star counts, with "Via Claude Code Desktop" noted in the message.

push it to remote

- **Bash git push -u origin zealous-jemison**

Allow Claude to Run git push -u origin zealous-jemison?

Push branch to remote

git push -u origin zealous-jemison

Deny Esc

Always allow for project (local) ⌘ ↵

Allow once ↵

3 files changed, 100 insertions(+), 13 deletions(-)

- Bash git status

- └ On branch vigilant-feistel
nothing to commit, working tree clean

- Done! Committed with message:

Add animations to landing page

- Add fade-in-up, slide-in-left, and float animations
- Hero section animates in with staggered delays
- Browse Hooks button has pulsing glow effect
- Logo floats in header
- Hook cards animate in with staggered timing and lift on hover
- Section headers slide in from left

Via Claude Code Desktop

push it to remote

- Bash git push -u origin vigilant-feistel

Allow Claude to **Run** git push -u origin vigilant-feistel?

Push branch to remote

git push -u origin vigilant-feistel

Deny Esc

Always allow for project (local) ⌘ ↵

Allow once ↵

- Animated badge with live indicator ("Community-Powered Automation")
- Bold three-line headline with gradient text effect on "powerful hooks"
- Animated underline SVG that draws in
- Enhanced CTA buttons with hover glow effects and micro-interactions
- Stats section (50+ Hooks, 1.2k Downloads, 200+ Contributors)

Hero Visual (right side):

- Orbiting elements around a central hook icon
- Spinning rings with dashed borders
- Rotating icons (notification bell, checkmark, lightning bolt)
- Inner glow effects

Animations Added:

- pulse-slow , float , float-delayed - for orbs
- ping-slow , bounce-slow - for nodes
- spin-slow , reverse-spin - for orbiting elements
- dash , dash-delayed - for SVG connection lines
- fade-in , slide-up - for staggered content reveal
- gradient - for animated text gradient
- draw - for underline drawing effect

All changes have been committed and pushed to `claude/anthropic-hero-design-01Fo8dkJxcBUqWHyjhPfTHQS`.

claude/anthropic-hero-design-01Fo8dkJxcBUqWHyjhPfTHQS

Create PR

Open in CLI

claude/anthropic-hero-design-01Fo8dkJxcBUqWHyjhPfTHQS had recent pushes 1 minute ago

Compare & pull request

vigilant-stel had recent pushes 8 seconds ago

Compare & pull request

zealous-jemison had recent pushes 6 seconds ago

Compare & pull request

project/hookhub

12 Branches 0 Tags

Go to file

t

Add file

<> Code

This branch is 4 commits ahead of and 3 commits behind main .

Contribute

emarco177 and claude Apply Anthropic brand guidelines to HookHub

b51dc55 · 2 days ago 5 Commits

hookhub

Apply Anthropic brand guidelines to HookHub

2 days ago

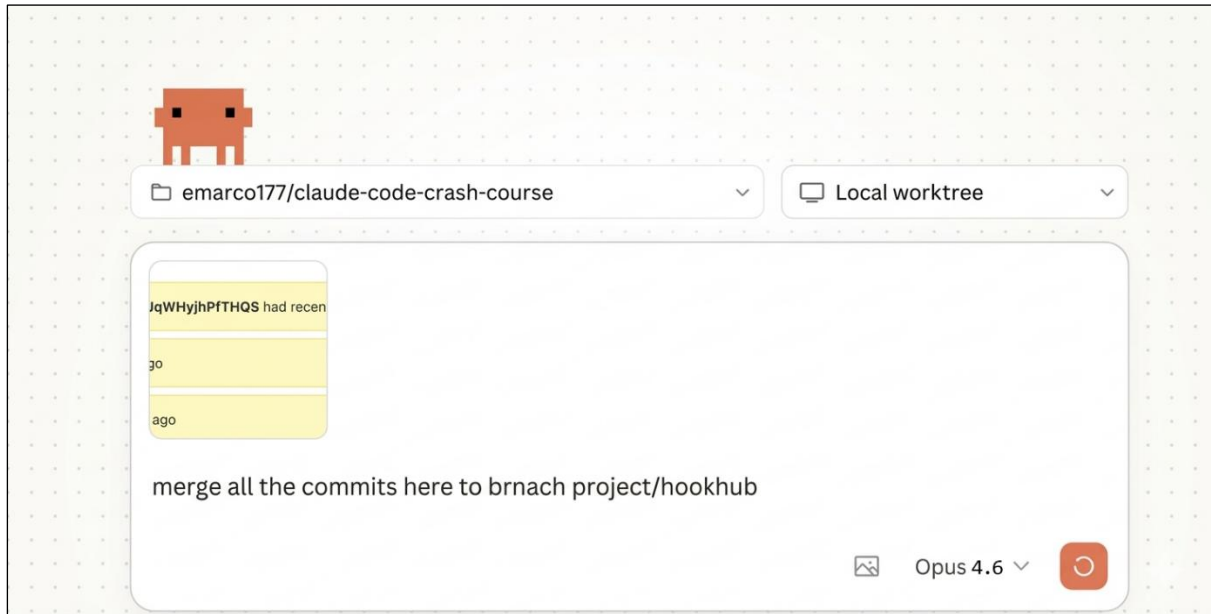
README



Add a README

Help people interested in this repository understand your project by adding a README.

Add a README



merge all the commits here to brnach project/hookhub

- I'll help you merge all these branches into `project/hookhub`. Let me first check the current state and then merge each branch.

- **Bash** `git fetch origin && git branch -a | head -30`

```
└─ remotes/origin/project/mcp
   remotes/origin/project/output-styles
   remotes/origin/project/skills
   ... +27 lines
```

- **Bash** `git checkout project/hookhub`

```
└─ Exit code 128
   fatal: 'project/hookhub' is already checked out at
   '/Users/edenmarco/GithubProjects/claude-code-crash-course'
```

- The branch is checked out in the main repository. Let me work around this:

- **Bash** `git fetch origin project/hookhub && git checkout -B project/hookhub-merge origin/project/hookhub`

```
└─ branch 'project/hookhub-merge' set up to track 'origin/project/hookhub'.
   From https://github.com/emarco177/claude-code-crash-course
    * branch          project/hookhub -> FETCH_HEAD
   Switched to a new branch 'project/hookhub-merge'
```

✦ Pondering...

```

claude/anthropic-hero-design branch"
└ Exit code 1
  CONFLICT (modify/delete): README.md deleted in HEAD and modified in
  origin/claude/anthropic-hero-design-01Fo8dkJxcBUqWHYjhPfTHQS. Version
  origin/claude/anthropic-hero-design-01Fo8dkJxcBUqWHYjhPfTHQS of README.md left in
  tree.
  Auto-merging hookhub/src/app/globals.css
  CONFLICT (add/add): Merge conflict in hookhub/src/app/globals.css
  Auto-merging hookhub/src/app/page.tsx
  CONFLICT (add/add): Merge conflict in hookhub/src/app/page.tsx
  Automatic merge failed; fix conflicts and then commit the result.

• Let me check the conflicts and resolve them:

• Bash git status --short
└ DU README.md
  AA hookhub/src/app/globals.css
  AA hookhub/src/app/page.tsx
  A static/banner.png

• Update Todos
└ ☐ Resolving merge conflicts from anthropic-hero-design branch
  ☐ Merge vigilant-feistel branch
  ☐ Merge zealous-jemison branch
  ☐ Push merged branch to origin/project/hookhub

```

* Inferring...

```

66      - <button className="px-6 py-3 border border-[var(--border)] hover:
67          border-[var(--slate-light)] text-[var(--foreground)] font-medium rounded-xl tra
68          nsition-colors duration-200">
69      -     Submit a Hook
        - </button>
        - =====

```

Show full diff (320 more lines)

- Now let me remove the README.md (we don't need it) and stage all the resolved files:
- Bash** git rm README.md && git add hookhub/src/app/globals.css hookhub/src/app/page.tsx static/banner.png
 - rm 'README.md'
- Bash** git commit -m "Merge claude/anthropic-hero-design branch"
 - [project/hookhub-merge dcf79dd] Merge claude/anthropic-hero-design branch
- Update Todos**
 - ☐ ~~Resolve merge conflicts from anthropic-hero-design branch~~
 - ☐ Merging vigilant-feistel branch
 - ☐ Merge zealous-jemison branch
 - ☐ Push merged branch to origin/project/hookhub
- Bash** git merge vigilant-feistel -m "Merge vigilant-feistel branch"

Allow Claude to Run git push origin project/hookhub-merge:project/hookhub?

Push merged branch to origin/project/hookhub

git push origin project/hookhub-merge:project/hookhub

Deny Esc Always allow for project (local) % % ↵ Allow once ↵

globals1/hookhub/src/app/globals.css



can u run it locally so i can test before merging?

- Sure! Let me start the development server for you.
- **Bash** `cd /Users/edenmarco/.claude-worktrees/claude-code-crash-course/suspicious-bassi/hookhub && npm run dev`
 - ↳ Command running in background with ID: 9a376f
- **BashOutput**
- Let me try with the full path:
- **Bash** `cd /Users/edenmarco/.claude-worktrees/claude-code-crash-course/suspicious-bassi/hookhub && /opt/homebrew/bin/npm run dev`
 - ↳ Command running in background with ID: 65f00f
- **BashOutput**
- **Bash** `which npm || which node`
 - ↳ Exit code 1
npm not found
node not found
- **Bash** `source ~/.zshrc 2>/dev/null || source ~/.bashrc 2>/dev/null; which npm`
 - ↳ (No content)

* Manifesting...

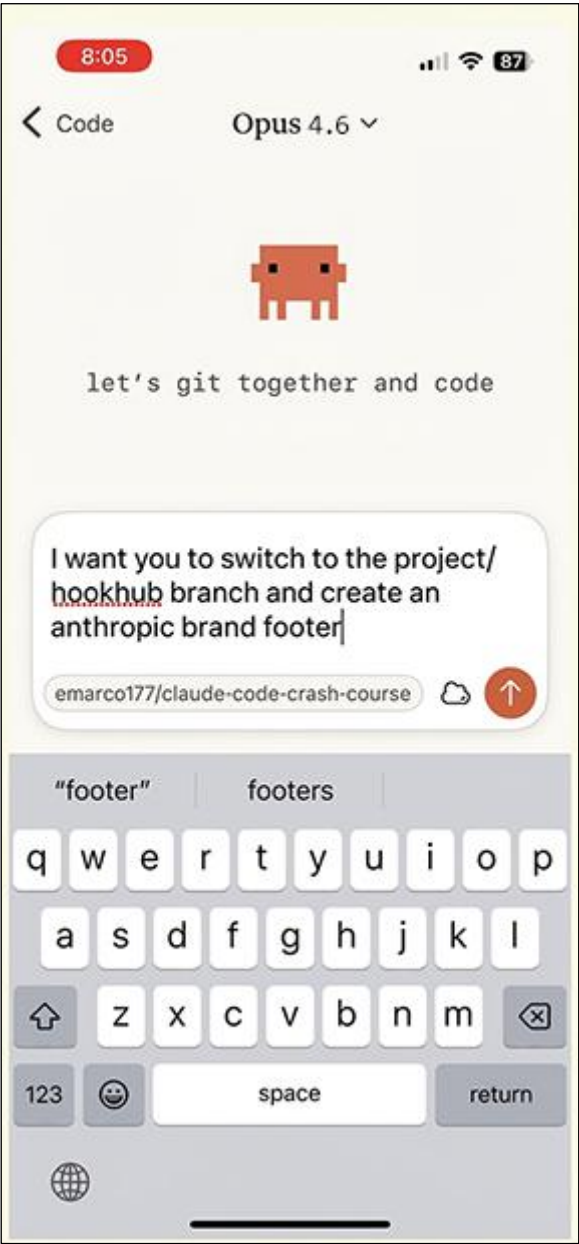
The image displays two side-by-side screenshots of the HookHub website, which is a platform for sharing and installing hooks to extend Claude Code's functionality.

Left Screenshot: The page is titled "Extend Claude Code with powerful hooks". It features a search bar, a "Browse Hooks" button, and a "Submit a Hook" button. Below this, there is a "Featured" section with two highlighted hooks:

- Claude Code Hooks Mastery** (FEATURED): Complete hook lifecycle implementation with deterministic control over Claude Code's behavior. It lists workflow hooks: `PRE_TOOL_USE`, `POST_TOOL_USE`, and `USER_PROMPT_SUBMIT`.
- Multi-Agent Observability** (FEATURED): Real-time tracking and monitoring of multi-agent workflows with detailed observability metrics. It lists monitoring hooks: `SUBAGENT_START`, `SUBAGENT_STOP`, and `SUBAGENT_STREAM`.

Right Screenshot: The page is titled "Supercharge Claude Code with powerful hooks". It features a search bar, a "Browse Hooks" button, and a "Submit a Hook" button. Below this, there are statistics: "50+ Hooks Available", "1.2k Downloads", and "200+ Contributors".

Commits		
🔍 project/hooks	👤 All users	📅 All time
Commits on Dec 1, 2025		
Merge zealous-jemison branch 👤 emarco177 committed 5 minutes ago	f961c60	📄 <>
Merge vigilant-feistel branch 👤 emarco177 committed 5 minutes ago	b6dab9c	📄 <>
Merge claude/anthropic-hero-design branch 👤 emarco177 committed 7 minutes ago	dcf79dd	📄 <>
Add top Claude Code hooks repositories to hooks list 👤 emarco177 and claude committed 13 minutes ago	d88a8b3	📄 <>
Add animations to landing page 👤 emarco177 and claude committed 13 minutes ago	b438f44	📄 <>
Add Anthropic-style hero design to HookHub 👤 claude committed 13 minutes ago	8f274de	📄 <>



8:05

87

< Create anthropic brand foot... ⋮

I'll switch to the project/hookhub branch and then explore the codebase to create an Anthropic brand footer.

```
Bash git fetch origin project/hookhub
&& git checkout project/hookhub
```

```
branch 'project/hookhub' set up to
track 'origin/project/hookhub'.
From http://127.0.0.1:37940/git/
emarco177/claude-code-crash-course
* branch                project/hookhub
```

```
Glob **/*
```

```
/home/user/claude-code-crash-
course/.git/info/exclude
/home/user/claude-code-crash-
course/.git/hooks/fsmonitor-
watchman.sample
```

```
Bash ls -la /home/user/claude-code-
crash-course
```

```
total 20
drwxr-xr-x 5 root root 4096 Dec  2
06:05 .
drwxr-xr-x 3 root root 4096 Dec  1
13:09 ..
```

+ -

Add feedback...

emarco177/claude-code-crash-course





- Add project README.md , CLAUDE.md , and specifications under memory/ .

- Misc:
 - Remove root README.md .

Written by [Cursor Bugbot](#) for commit [0a3bba0](#) . This will update automatically on new commits. Configure [here](#) .





 Add Anthropic brand footer component ...

Verified

0a3bba0



 emarco177 changed the base branch from `main` to `project/hookhub` 1 minute ago



emarko177 commented now

[View reviewed changes](#)

emarko177 left a comment

Owner

Author

...

@claude review this PR





No conflicts with base branch
Merging can be performed automatically.

Merge pull request

You can also merge this with the command line. [View command line instructions.](#)

Still in progress? [Convert to draft](#)

Add a comment

Write

Preview

H

B

I

≡

<>

↗

|≡

|≡

≡

🔗

@

🗨

↶

👍

Add your comment here...

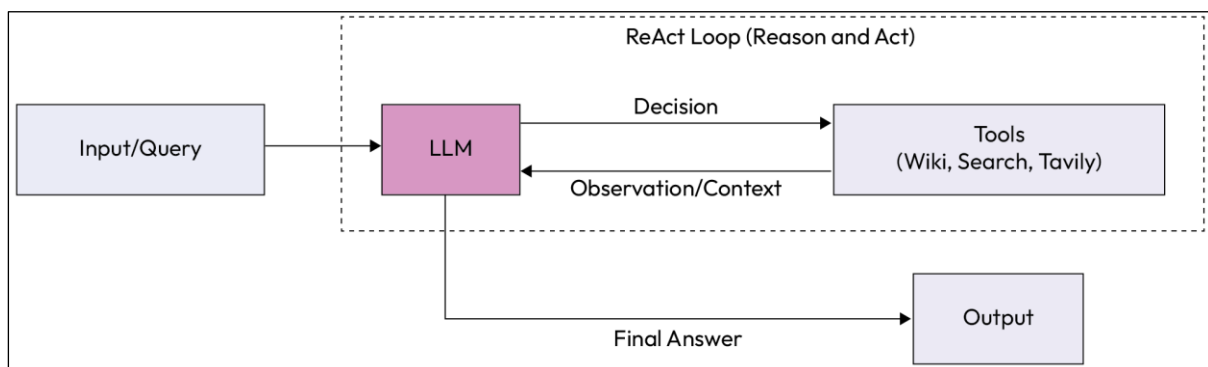
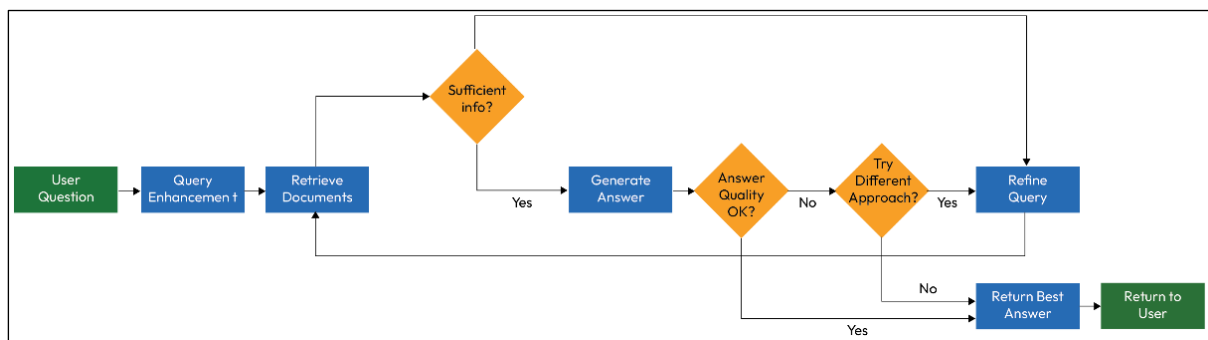
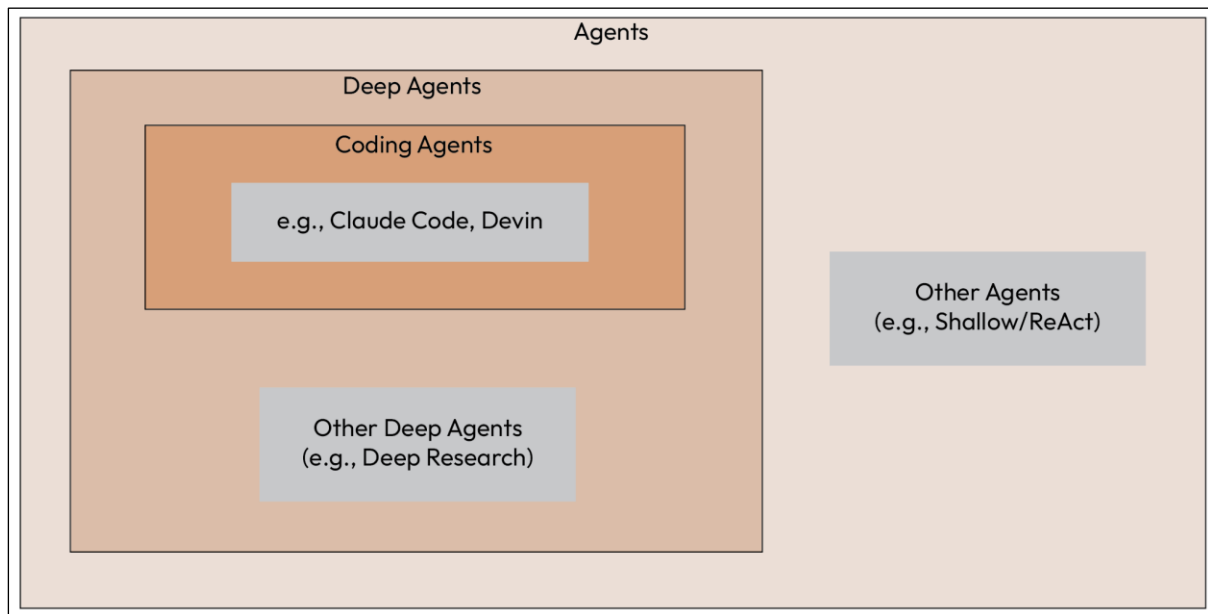
 Markdown is supported

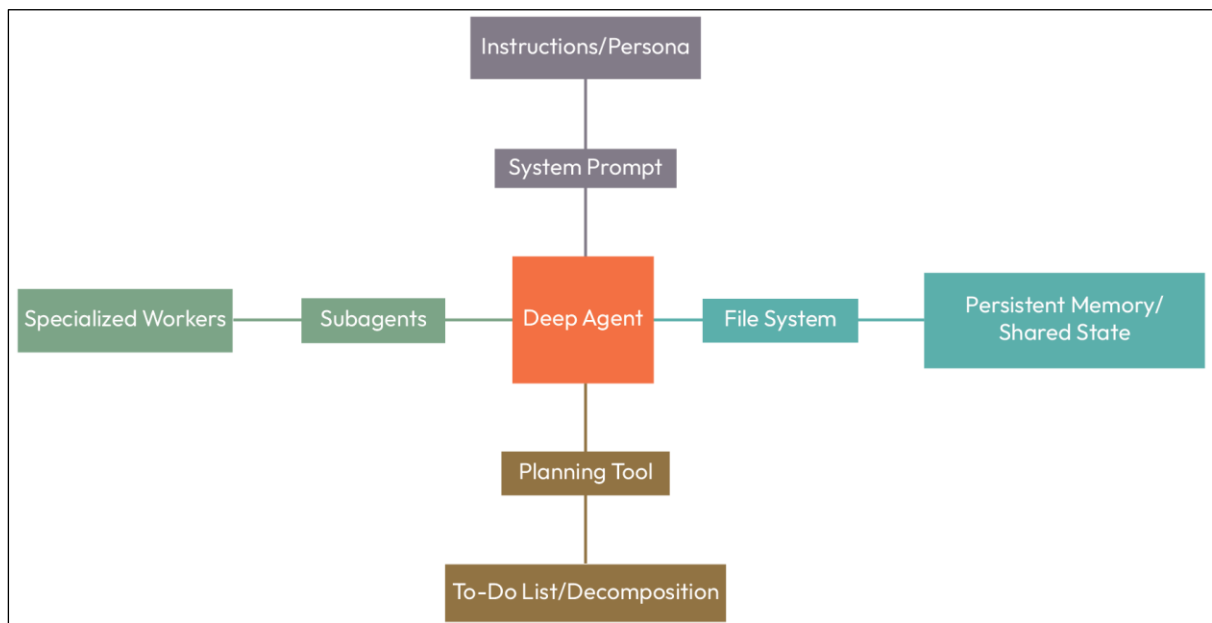
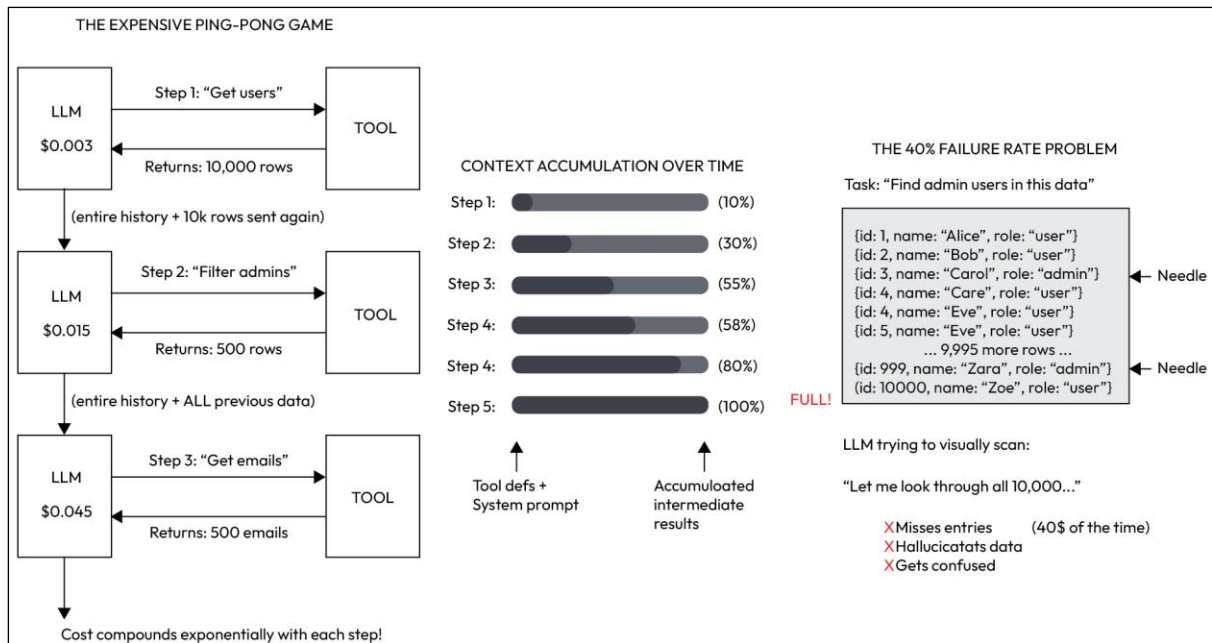
 Paste, drop, or click to add files

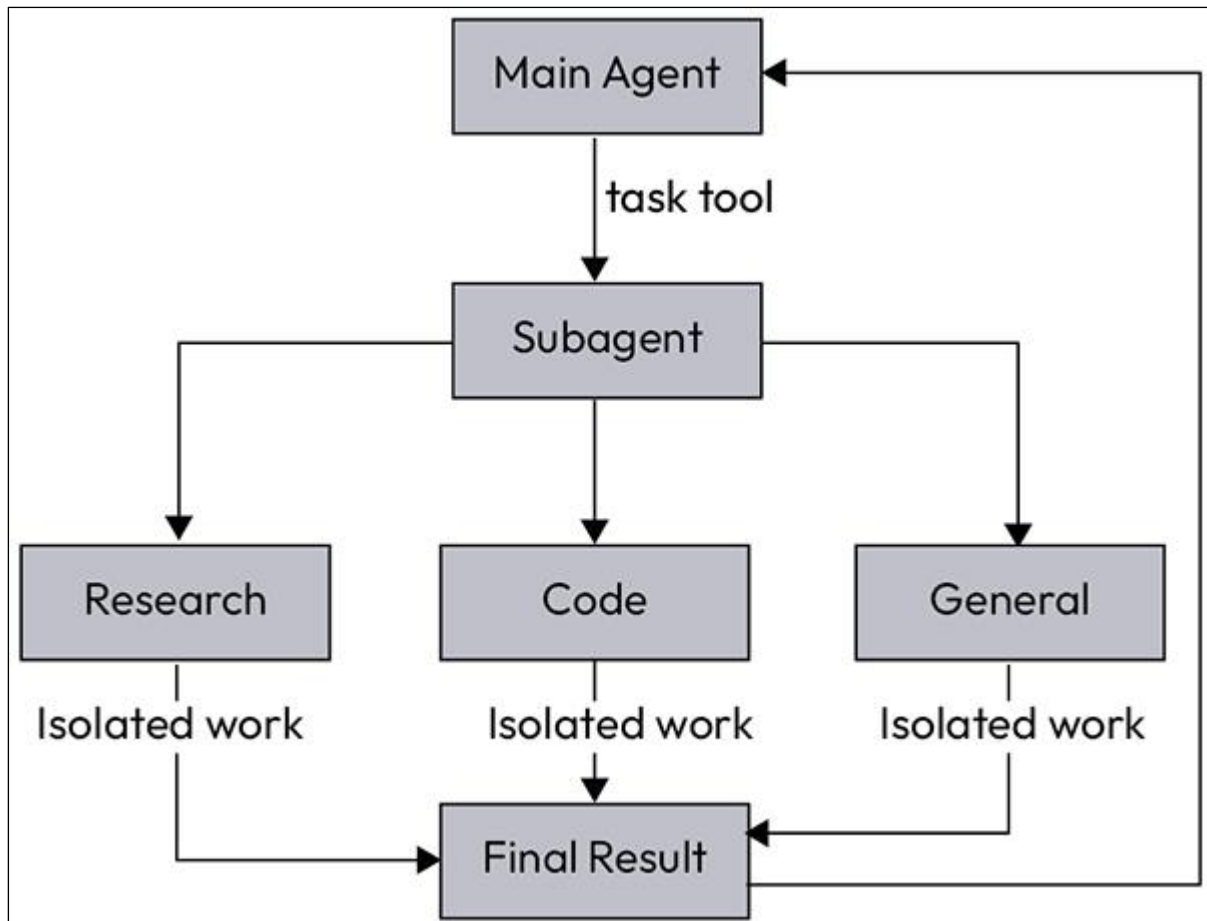
 Close pull request

Comment

Chapter 11: Understanding Deep Agents







```
• I'll spawn a sub-agent now to demonstrate hierarchical delegation:

• Explore(Find authentication patterns) Haiku 4.5
  L /Users/edenmarco/.claude-worktrees/claude-code-crash-course/vigilant-feistel/hookhub/node_modules/queue-microtask/index.js
  Waiting_d /Users/edenmarco -type f \( -name "*.js" -o -name "*.ts" -o -name "*.jsx" -o -name "*.tsx" \) ! -path "*/node_modules/*" 2>/dev/null)
  Search(pattern: "authlogin|jwt|session|password|token|credential", glob: "**/*.{js,ts,jsx,tsx,py,java}")
  +10 more tool uses (ctrl+o to expand)
  ctrl+b to run in background

• Implementing auth middleware_ (esc to interrupt · ctrl+t to hide todos)
  L ■ Research-existing-authentication-patterns-in-codebase
    ■ Design-JWT-token-structure-and-refresh-flow
      □ Implement auth middleware
      □ Create login/logout API endpoints
      □ Add password hashing with bcrypt
      □ Write integration tests for auth flow
      □ Update API documentation
```



Claude Code v2.1.37

Opus 4.6 • Claude Max

~/Desktop

Opus 4.6 is here - Most capable for ambitious work

> which tools do u have

- Here's a summary of the tools I have available:

File Operations

- Read - Read file contents (including images, PDFs, notebooks)
- Write - Create/overwrite files
- Edit - Make precise string replacements in files
- Glob - Find files by pattern (e.g., `**/*.ts`)
- Grep - Search file contents with regex
- NotebookEdit - Edit Jupyter notebook cells

All context available to the agent

What the agent
needs to answer
question X

What the agent
selects and
pulls into its
context window

Box 1

Box 2

Box 3